The Amiga 3000+

System Specification

An enhanced Amiga 3000 family computer

Document Revision 0.6

1991 DevCon Release

by
Dave Haynie

July 17, 1991

IMPORTANT INFORMATION

This Document Contains Preliminary Information

The Amiga 3000+ computer described in this document is currently a work in progress. While this is an honest attempt to specify the A3000+ computer, it is still very preliminary in nature and subject to possible errors and omissions. Additionally, the specification of the A3000+ itself is still subject to possible change at this stage in the design cycle.

Commodore Technology reserves the right to correct any mistake, error, omission, or viscious lie. Corrections will be published as updates to this document, which will be released as necessary in as developer-friendly a manner as possible. Revisions will be tracked via the revision number that appears on the front cover.

This document does not constitute a guarantee that Commodore will build or offer the A3000+, as described herein or in any other form, for sale at any given time, place, cost, etc. Nor is this a guarantee that Commodore will ever release anything called "Amiga 3000+". This is simply the design and working name for a prototype next generation Amiga 3000 class computer as of the cover date.

Most of the text in Chapter 3, and the Lisa chip itself, were created by Bob Raible. Without Bob, and the other "AA" guys, Bill Thomas and Chingtao Shen, there would be no A3000+.

TABLE OF CONTENTS

TABLES AND FIGURES

CHAPTER 1

INTRODUCTION

Thou shalt not bring into the world knowledge and
devices that will harm mankind.
-Herbert A. Simon

The Amiga 3000+ computer is a second generation Amiga 3000 class computer. It shares many of the Amiga 3000 system's features, while adding a significant number of enhancements, designed to increase the Amiga line's support of more advanced video and audio playback and processing. It is intended to be upward compatible with the Amiga 3000, and shares most mechanical specifications with the Amiga 3000 as well.

This document is intended to collect, in one single place, all of the hardware details of the Amiga 3000+. Obviously, it won't reach this goal completely. There is far too much information pertinent to the A3000+ to reproduce in a single document. Therefore, where appropriate, additional documents are referenced, leaving the bulk of the space available for material specific to the A3000+. This document should be all that's necessary, for example, to port a new operating system to the A3000+, assuming one obtains other generic materials, such as the microprocessor and expansion bus specifications. It also collects all A3000+ connector pinouts and I/O register definitions in a single place, though both of these items may be better described in other documents.

1.1 Basic Description

The Amiga 3000+ takes the best features of the Amiga 3000, like 32-bit architecture, video compatibility, expandability, and adds new features to better suit current and future graphics,

sound, and video applications. The complete feature list is, in essence, this document. The basic highlights, though, are as follows:

- 25MHz MC68030/MC68882 Main Processor
- 50MHz ATT DSP3210 Coprocessor
- Amiga "Pandora" graphics chip set
- Improved 32-bit DRAM and SCSI DMA controllers
- Enhanced video expansion slot
- New audio I/O ports
- Support for 4MB ROM
- Built-in telephone line interface
- Built-in 2.5 Mbit/second RS-485 network port

The basic system architecture is shown in *Figure 1-1*. The organization of the A3000+ hardware is very similar to that of the A3000. The central processor is a Motorola 68030 clocked at 25MHz, with 68882 floating point coprocessor also at 25MHz. The main motherboard controller is the Gary+ chip, a custom gate array. This generates chips selects for and timing for motherboard slave resources other than Fast RAM. On the A3000+, these resources include the Amiga "Pandora" chip set (Alice, Lisa, Kelly, and Paula), up to 4MB of ROM, two 8520 Complex Interface Adaptor chips, a Standard Microsystems COM20020 network interface chip, and a custom Amiga DMA controller for SCSI bus. On the A3000+, the DMAC also contains registers for coprocessor control and the SSPB interface, a Synchronous Serial Peripheral Bus used for battery backed RAM/Clock, Audio processor, and Video Slot expansion.

To additional motherboard resources manage their own control functions. The RAMSEY chip controls up to 16MB of memory on the motherboard, with facility for burst and page-detection modes when used with static column DRAM. The RAMSEY device also acts as an address generator for the DMAC during SCSI DMA transfers. The Buster chip controls the standard Amiga expansion bus, suppoting both Zorro II and Zorro III procotols. It manages bus arbitration for both expansion, motherboard, and coprocessor slot resources.

In addition to the 68030 system, the Amiga 3000+ has an on-board DSP coprocessor, the ATT DSP3210 clocked at 50MHz. This processor has direct memory access to all 32 bit motherboard resources, just as the 68030. In addition, it has a private high speed serial bus with DMA access to 3210 internal memory. The A3000+ provides two serial bus devices, a telephone line interface CODEC and a 16 bit stereo audio CODEC. The DSP expansion connector provides two additional channels on the serial bus, to support the additional DSP peripherals.

1.2 Memory Organization

All Amigs 3000+ memory and I/O resources are memory mapped. The A3000+ memory map is illustrated in *Figure 1-2*. This is a superset of the A3000 memory map, with allocations for the new A3000+ resources and definitions for various extensions to this basic memory map to handle possible future enhancements
There are a few memory aliases present in this map, to support backward compatibility with the

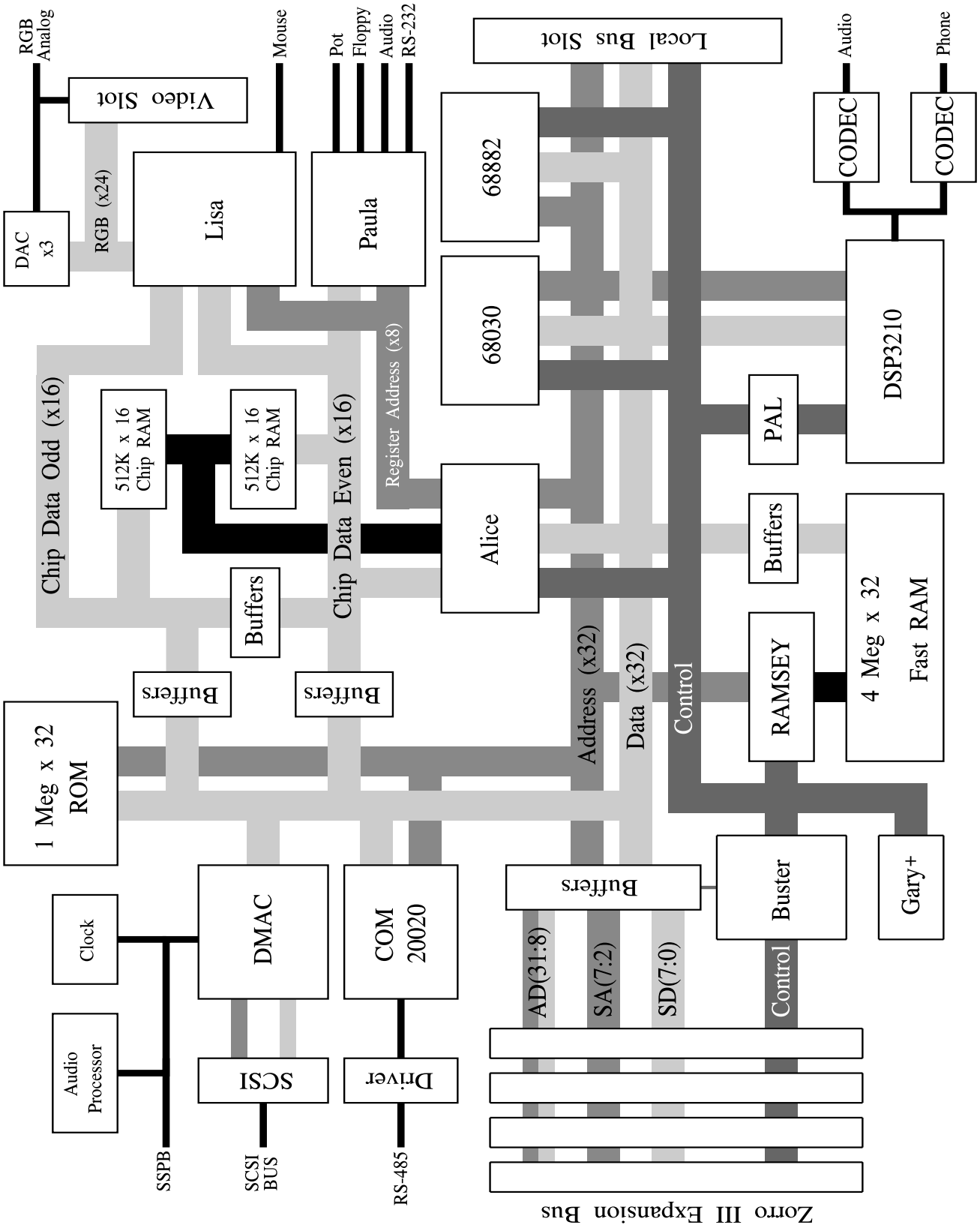


Figure 1-1: Amiga 3000+ System Structure

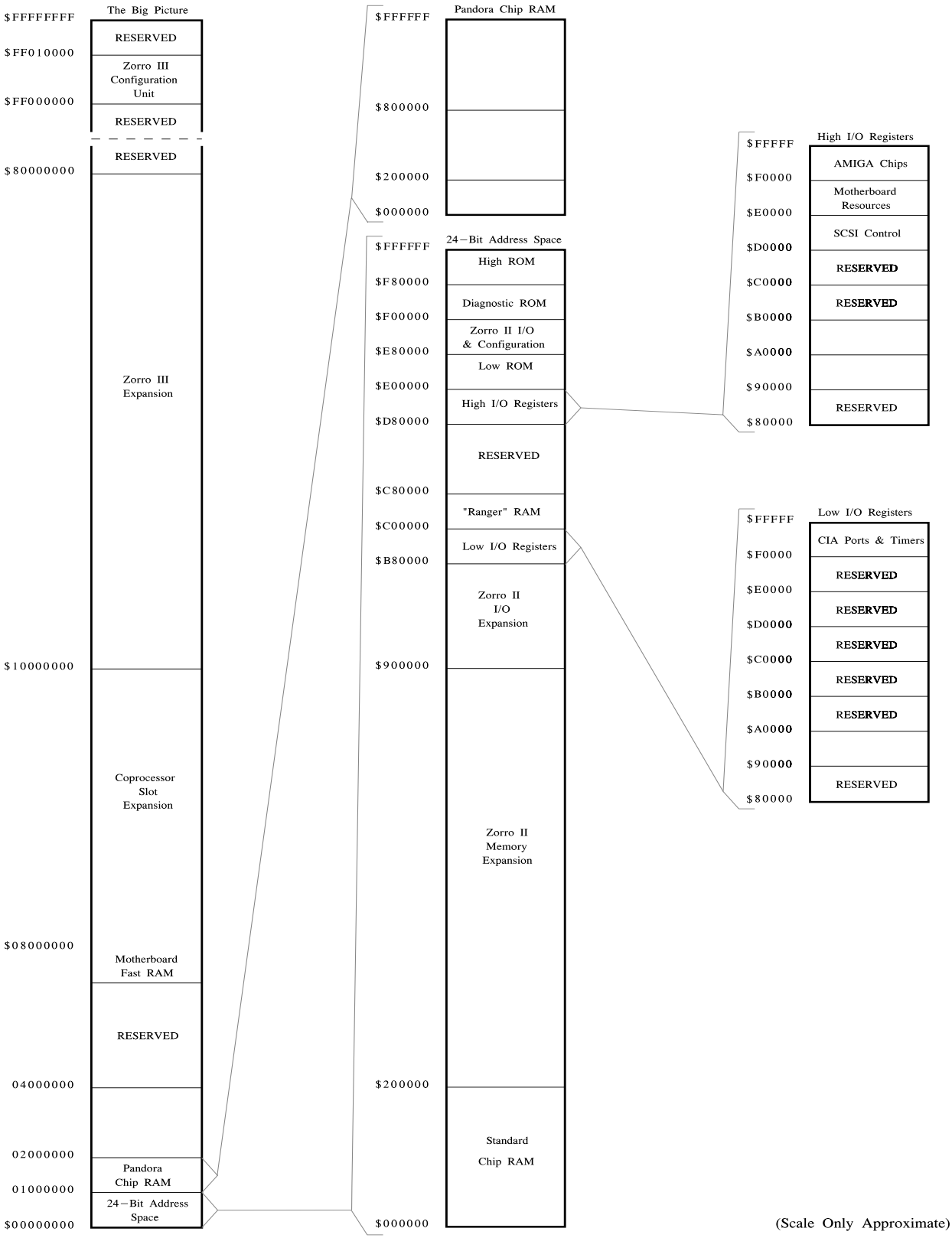


Figure 1-2: Amiga 3000+ Memory Map

A3000 and earlier systems. Chip RAM, for example, is mapped as in earlier systems at $00000000-$001FFFFF. Since the orginal Amiga architecture supported only 2MB of Chip RAM, though, any additional Chip memory will have to be mapped non-contiguously. The A3000+ solution to this is to map the memory at $00000000 to the base of a new region, starting at $01000000. The A3000+ architecture physically supports 8MB of Chip RAM, so should an 8MB Alice be designed, the additional 6MB winds up in the $01200000-$017FFFFF range. The next 8MB has been reserved for use as Chip RAM in future systems.

Similarly, the A3000 used up all the reserved ROM space in the original Amiga memory map, the range from $00F80000-$00FFFFFF, for the 512K space used by AmigaOS 2.0. To support at least 1MB of ROM on machines with 24 bit addressing, the previously unused 512K from $00E00000-$00EFFFFF. To support additional ROM in an upward compatible fashion, the A3000+ addresses 4MB of ROM at $03C00000-$03FFFFFF. The 512K chunk at $00F80000 is also mapped at $03F80000. and the 512K chunk at $00E00000 is also mapped at $003E0000.

Some details of the A3000+ memory organization are too small to show up on this chart. The Gary+ has a number of important control registers, which are described in the next chapter. The DMAC, along with its normal DMA controller functions, also has the Coprocessor Control and SSPB registers mapped in its space. These are also covered in the next chapter. While not shown in the memory map, the DSP3210 has some internal resources that overlay the A3000+ memory as mapped by the 68030. These will be covered in the DSP3210 chapter.

CHAPTER 2

MAIN SYSTEM CONTROLS

We're definitely coming to the post-IBM era.
-Joel Tessler

There are lots of function in the Amiga 3000+, and not surprisingly, lots of control registers associated with these features. Theese control registers are physically located in the various system chips.

2.1 Gary+

The Gary+ chip is essentially the motherboard controller in the Amiga 3000+. It performs the following functions:

- Chip select and sometimes timing for Alice, the two 8520s, the DMAC, the ROM, the 68882, the COM20020, and the Local Bus Slot select.
- Chip RAM buffer and CPU byte access control.
- Genlock/internal clock multiplexing
- Reset control, including powerup detect, keyboard reset with disable, and DSP reset.
- Fast interrupt disable control.
- Bus timeout and error control.
- High order address generation for Zorro II DMA.

The chip selects should be pretty self-explanatory. The Gary+ design insures that each chips gets the length and bus size it requires. The ROM cycle is programmable via a PCB jumper to either

160ns or 280ns. Access to Chip RAM is done, as in other Amiga system, according to what the Chip RAM controller, Alice in this case, is doing. It is still possible to set up bandwidth intensive display and blitter functions with Alice, so the Chip RAM access is alway potentially variable.

Gary+ has a number of programmable control registers. These registers are located on byte boundaries, in Supervisor data space, and each contains one significant bit, which will be D_7 in the access byte. The registers are shown in *Table 2-1*.

Register	Bit	Name	Function
$00DE0000	7	TIMEOUT	Controls bus timeout
$00DE0001	7	TOENB*	Time out enable
$00DE0002	7	COLDSTART	Indicates cold powerup
$00DE1000	7	DSPRST*	DSP reset line
$00DE1001	7	KBRSTEN	Keyboard reset control
$00DE1002	7	GARYID	Serial ID code

Table 2-1: Gary+ Registers

2.1.1 Bus Timeout Support

The Gary+ timeout support is necessary to trap processor access to unmapped memory resources. In early Amigas, the motherboard controller would simply run zero wait state cycles for all memory accesses unless instruction to add waits for Chip bus or expansion resources. This policy has two problems. First of all, it makes detecting access to non-existant memory very difficult. Secondly, it pretty much requires one central agency to do all of the cycle timing in the system. Being a more sophisticated Amiga system, the A3000+ really needs the ability to trap random addressing in hardware. And cycle termination isn't central; it comes from several different places, such as Gary+ itself, RAMSEY, Buster, the 68882, and a possible slave device in the Local bus slot.

At power up, the TIMEOUT and TOENB* registers are both zero, which indicates that the default timeout is selected. Default timeout yields an automatic 32 bit asynchronous cycle termination (via /$DSACK_0$ and /$DSACK_1$) after 9µS, effectively ignoring the access. This mode is used during power up, since the OS polls certain memory locations that may not respond, and we don't want powerup to take all day. When a one is written to TIMEOUT, the detectable timeout mode is set. This causes a termination via /BERR after 250 mS. This is the normal operating mode of A3000 class systems under AmigaOS 2.0. For special purposes, timeout may be turned off altogether. This is done by writing a one to the TOENB* register. Also, since the Chip bus may lock the CPU out for extended periods of time, all timeout logic is disabled during an access to any Chip bus resource.

2.1.2 Reset Control

Gary+ has several registers associated with system reset. The COLDSTART register detets a power up condition, so that software can distinguish cold boots from warm boots. This register will be set high on power up only; once written low, it will stay low until the next power cycle. The DSPRST* line goes low on reset and stays low until a one is written to it. This allows the 68030 to get things in order before turning on the DSP3210. The KBRSTEN bit powers up low, which allows keyboard resets to generate system reset just like in any other Amiga system. If this bit is set high, however, the keyboard reset will be ignored.

2.1.3 Gary+ ID

The GARYID register is used to get the 8 bit Gary+ ID, which identifys the type of Gary and the revision level of that type. To initiate the ID code stream, a zero is written to GARYID. The bits of the Gary ID code are then read from GARY+ in eight reads of the GARYID register. The code is extracted MSB first. The first four bits of the code indicate the type of system controller, the last four bits indicate the revision level. This register is constructed in such a way that this same technique works on the Amiga 3000. The A3000's Gary returns the code $00, the Gary+ returns the code $90.

Register	Bit	Name	Function
$00DE0003	7	TEST	Used for production test
	6,5	REFRESH RATE	Timing for refresh
	4	RAMWIDTH	Set up for x4 or x1 RAM
	3	RAMSIZE	1MB or 4MB density
	2	WRAP	Enable burst wrapping
	1	BURST	Run 68030 burst cycles
	0	PAGE DETECT	Use page-detect logic
$00DE0043	7-0	VERSION	RAMSEY chip version

Table 2-2:RAMSEY Registers

2.2 RAMSEY

The A3000+ RAM controller is an improved version of the A3000 RAMSEY chip. The RAMSEY chip directly manages 16MB of 32 bit wide Fast RAM. A number of the RAMSEY features can be tuned, depending on the memory connected. The RAMSEY registers, all 8-bit registers located in Supervisor Data space, are show in *Table 2-2*. Writes to $00DE0003 don't actually take effect until the next refresh cycle. A read loop on the changed bit can be used if it necessary to wait for a change. RAMSEY controlled Fast RAM is located from $07FFFFFF building down toward $00700000, as far as the populated memory will take it.

2.2.1 Memory Configuration

The preferred A3000+ configuration is with static column DRAM in the 256Kx4 or 1MBx4 packages. RAMWIDTH defaults to one, which sets RAMSEY up for operation properly as far as the A3000+ is concerned. With RAMWIDTH set to zero, RAMSEY will drive 1MBx1 DRAM, though the A3000+ motherboard can't physically accomodate these parts. RAMSIZE is set by J852 on the A3000+ motherboard, it reads low for 256Kx4, high for 1MBx4. The RAMSIZE register can be written to, though there's generally no need for that.

DRAM Refresh is programmed by the REFRESH RATE bits, according to the timing shown in *Table 2-3*. The refresh time may have to be adjusted for the memory mode in effect. In basic access mode, the DRAM need one refresh cycle every 15.625µS or more. However, during a static column cycle resulting from page-detect mode, the DRAM need a refresh cycle every 10 µS or more. Any program modifying the OS or powerup settings of these bits needs to take this into acount.

RATE	# Clocks	Interval (µS)
00	154	6.16
01	238	9.52
10	380	15.2
11	N/A	No Refresh

Table 2-3: Refresh Timing

2.2.2 Static Column Modes

On powerup, RAMSEY runs a five clock cycle to DRAM. This cycle will work with either page mode or static column DRAM rated for 80ns operation. There are two enhanced modes that take advantage of static column memory to speed things up. The first of these is the burst mode, which is set by setting the BURST bit high. This causes RAMSEY to assert /CBREQ when accessed and run a 68030 style burst cycle when /CBACK is asserted by the current bus master. Burst cycles take two clocks each. The burst cycle operation is modified by the WRAP bit. The 68030 burst cycle always starts on a longword that the 68030 wants, and in the process gets the other three in the natural quadlongword defined by the first access. This means that burst wraps around behind the first longword when that longword isn't first in the quadlongword. With BURST set to one, natural 68030 burst takes place. With BURST set to zero, the burst cycle won't wrap backwards, but instead terminate with less than four longwords fetched. Note that the 68040 doesn't support aborted bursts, so WRAP much be one for most 68040 systems.

The other enhanced RAMSEY mode is called page-detect mode. When RAM is first accessed, RAS is held low after the cycle completes. This leaves the current RAM page "open", and the RAM bank will act like static RAM while accesses continue on this page. While accesses continue on-page, access time is driven by column rather than the longer row access time. Memory cycles complete in three clocks while on-page. Comparators in RAMSEY monitor the

new row addresses for every new cycle until a refresh comes along, which can be for up to 10µS The downside of this mechanism is that, when the page detector finds a page miss, the currently open page must be closed properly before a new one can be opened. This takes seven clocks, two to close the page, five to normally open the new one. This mode tends to work very well when memory accesses tend to be linear. It helps considerably during access by the DMAC or DSP3210, both of which have very linear access patterns. Under normal 68030 and AmigaOS use, this mode is sometimes a win, sometimes a loss.

Note that, due to a chip bug, original A3000 versions of RAMSEY (version code $0D) don't correctly support this mode. The A3000+ RAMSEY supports this mode properly. Any program messing with RAMSEY control registers should check for a version $0E or later RAMSEY before enabling this feature.

2.2.3 Page Mode/Static Column Detection

RAMSEY was originally designed to use static column memory exclusively. Because of these, there's no explicit support of page mode DRAM, though of course, page mode memory will work as long as no special features are enabled. The trick is detecting that page mode memory is in the A3000+ in the first place. The method for checking is as follows:

1) Disable all interrupts
2) Turn page detect mode on via the PAGE bit (be sure to wait until it takes effect).
3) Write $5AC35AC3, $AC35AC35, $C35AC35A, $35AC35AC to four consecutive longwords in a quadlongword.
4) Turn page detect mode off via the PAGE bit (again, wait for it to take effect).
5) Compare what's now in the selected quadlongword with what was written. If the data is correct, that bank has all static column DRAM in it.
6) Repeat steps 2 through 5 for each bank of memory.
7) Re-enable interrupts. If any bank contains page mode DRAM, none of the static column features can be used.

2.3 COM20020

The COM20020 Universal Local Area Network Controller is the engine behind the low cost RS-485 network in the A3000+. This device implements the physical layer of a 2.5 Mbps deterministic, token passing network interface. The device has 2K of on-chip dual port RAM for buffering of packets in both directions

The A3000+ configures the COM20020 in a bussed network configuration, based on the RS-485 twisted pair bus standard. The A3000+ uses RCA-type phono jacks for the physical connection. The connector is self-terminating; when a jack is plugged in, the terminator is removed from the bus. In this way, it's guaranteed that termination is placed at the end of the bus, and only at the end of the bus.

The COM20020 is fully described in the document COM20020 Universal Local Area Network Controller (ULANC), from Standard Microsystem Corporation, Component Products Division. The part is mapped in the A3000 system as a 32 bit device, located at $00D90000. Since the device has an eight bit data bus, each COM20020 register occupies only the first eight bits of each consecutive longword based location. The basic register map is given in *Table 2-4*.

Address	Read Function	Write Function
$00D90000	Status	Interrupt Mask
$00D90004	Diagnostic Status	Command
$00D90008	Address Pointer High	Address Pointer High
$00D9000C	Address Pointer Low	Address Pointer Low
$00D90010	Data	Data
$00D90014	Reserved	Reserved
$00D90018	Configuration	Configuration
$00D9001C	TentID/NodeID/Setup	TentID/NodeID/Setup

Table 2-4: COM20020 Register Mapping

2.4 DMAC

The DMAC is an improved version of the A3000 DMAC. This chip's main function is to act as a data transfer unit to the Western Digital 33C93 SCSI controller. The A3000+ version of this part has a FIFO increased to eight longwords deep, and it can now run as fast as the 68030 to 32 bit memory. It can also handle word-aligned DMA transfers, where the earlier device required longword aligned buffers. The new DMAC is designed to be compatible with the A3000 and A3000 software, so some bit assignments and features may seem strange.

Due to pin count restrictions, most DMAC control registers are not mapped strictly according to 68030 bus rules. All DMAC registers are accessed as longword-wide registers. However, they will always behave as whole longwords, even if accessed via word or byte instructions; it is impossible to independently access individual words or bytes within these registers. Therefore, any writes to DMAC registers must write all significant bits, and any reads from DMAC registers will appear to read the entire register, even if the machine op-code would claim otherwise.

2.4.1 DMA and SCSI Control

The DMAC and WD 33C93 SCSI controller chips work together as a team for most kinds of SCSI transfers. The WD 33C93 registers are actually mapped into the DMAC register space, as shown in *Figure 2-5*. Registers labelled "Read" are read-only registers, and "Write" are write-only. Those labelled "R/W" can be read or writtern. "Strobe" registers will cause a

specific action to take place when read or written, regardless of the data value involved. The WD3393 registers are mapped a little strangely. There are two WD33C93 registers, each of which is a byte-wide register. These were mapped in the A3000 based on 68030 behavior, rather than 68030 specificatios, so properly designed 68040 cards could not access these registers as bytes. Allowing access to SASR_L solved this for the 68040, but due to a cache bug in the 68030, this caused a problem correctable only with the MMU. The mapping in the enhanced DMAC provides byte-wide mapping that works for 68030 or 68040.

Register	Name	Type	Function
$00DD0004	WTC	R/W	Word Transfer Count (obs)
$00DD0008	CONTR	R/W	Control Register
$00DD000C	ACR	R/W	Address Control Register
$00DD0010	ST_DMA	Strobe	Start DMA
$00DD0014	FLUSH	Strobe	Flush FIFO
$00DD0018	CLR_INT	Strobe	Clear Interrupts
$00DD001C	ISTR	Read	Interrupt Status Register
$00DD003C	SP_DMA	Strobe	Stop DMA
$00DD0040	SASR_L	Write	WD33C93 SASR long (obs)
$00DD0041	SASR_B	Read	WD33C93 SASR byte (obs)
$00DD0047	SCMD_B	R/W	WD33C93 SCMD byte
$00DD0049	SASR_B	R/W	WD33C93 SASR byte

Table 2-5: DMAC/SCSI Registers

The WTC register is considered obselete, and is not actually supported in the A3000+ DMAC. It can be used to determine if new or old style DMAC is in a system, however. Bit two of the register is fixed at zero in the new part, but is a read/write bit in the old part. So writing a longword with bit two set, then reading it back, will determine which part is installed in a system. All A3000+ systems will have the new part, though A3000 systems may have either.

The DMAC control register, CONTR, contains four significant bits in a 32-bit word, as shown in *Table 2-6*. The DMAENA bit, reads high end DMA is enabled, low when DMA is disabled. Reset or the SP_DMA strobe cause this to go low, while the ST_DMA strobe causes it to go high. The PREST bit will reset the WD33C93 if a high is written to it, while low lets the WD33C93 operate normally. The INTENA enables the generation of a system interrupt by the DMA controller. Finally, the DMADIR bit indicates the direction of data transfer for a DMA operation. A high written here causes a data transfer from Amiga memory to the SCSI bus, while a low here causes data transfer to go from the SCSI bus to Amiga memory.

The ACR indicates the starting byte in the 32 bit address space for a DMA transfer. Note that the ACR, physically contained in the RAMSEY chip, will actually round any value written to it down to an even word, since the DMAC doesn’t support odd-byte aligned transfers. This

register must be updated for every new DMA transfer. Note that the length of the transfer can actually be an odd number of bytes, as explained in the discussion of the FLUSH strobe.

The ST_DMA strobe starts the DMA transfer. Once the ACR and CONTR registers are properly set for a transfer, a read or write of this strobe causes the transfer to take place. The ACR must be reset for every new DMA transfer. The SP_DMA strobe causes the current DMA to unconditionally abort, and all internal DMA related registers to be reset. This location must be strobed prior to setting up a new DMA transfer. The FLUSH strobe causes the FIFO to be flushed in the case of a transfer from SCSI to main memory. This location must be strobed after the WD33C93 indicates the end of SCSI activity. If the last element of the FIFO contains one or two bytes, a word is written; if three or four bytes, a longword is written. A strobe of flush causes immediate DMA activity; the FIFO will be flushed on the instruction following the strobe, from the CPU's point of view.

Register	Bit	Name	Function
CONTR	8	DMAENA	DMA enabled
	4	PREST	WD33C93 reset
	2	INTENA	Interrupt enable
	1	DMADIR	DMA direction
ISTR	7	INT_F	Interrupt follow
	6	INT_S	Interrupt SCSI
	5	E_INT	End of process
	4	INT_P	Interrupt pending
	1	FF	FIFO Full
	0	FE	FIFO Empty

Table 2-6:Control/Interrupt Bit Assignments

The CLR_INT strobe clears all interrupts registered by ISTR, and negates the DMAC's interrupt output line. The ISTR register is responsible for recording interrupts, as shown in *Table 2-6*. Bits INT_F, INT_S, and E_INT each reflect the status of the WD33C93 interrupt line, which is an active high interrupt. The INT_P bit is low if INTENA is cleared, otherwise it indicates the status of the WD33C93 line, and will of course also indicate that an interrupt has actually be sent to the system by the DMAC. The FF bit goes high when the FIFO is full, and the FE bit goes high when the FIFO contains less than one longword.

The SASR and SCMD registers are part of the WD33C93 specification. They are fully described in the Western Digital publications "WD33C93A SCSI Bus Interface Controller" for the WD33C93A, and "WD33C93B (SBIC) Enhanced SCSI Bus Interface Controller" for the WD33C93B. The current intent is to use the latter part in the A3000+, though that depends on availability from Western Digital.

2.4.2 Coprocessor Interface

The coprocessor interface register is located at $00DD0050, with data direction register at $00DD0054. This register is used for signalling between the 68030, the DSP3210, and optionally, a Local Bus Slot device. The A3000+ system assignments for this port are given in *Table 2-7*. The 32-bit register contains eight significant bits, all of which are reset to inputs (DDR is set to $xxxxxxFF) on reset. Bits seven and six are currently unassigned.

The bit assignments start with the five and four, which are for coprocessor slot communications. The CI2P* bit is earmarked as an input from the coprocessor slot. The PI2C* bit is earmarked as an output to the coprocessor slot. These bits have no pre-defined definition, but instead are for the use of the private signalling protocol that may be desired between a coprocessor slot device and the main system.

Bit	Name	DDR	Function
7,6	N/A	I	Reserved
5	CI2P*	I	Coprocessor signal to processor
4	PI2C*	O	Processor signal to coprocessor
3	INT6*	I	DSP caused level 6 system interrupt
2	INT2*	I	DSP caused level 2 system interrupt
1	DSP1*	O	Cause DSP level 1 interrupt
0	DSP0*	O	Cause DSP level 0 interrupt

Table 2-7: Coprocessor Control Bit Assignments

The rest of the coprocessor interface register is concerned with the motherboard's DSP3210 coprocessor. The DSP3210 can generate level 6 or level 2 interrupts, in the traditional Amiga shared interrupt protocol. INT6* and INT2* should be read, respectively, by DSP interrupt handlers to determine if the DSP was in fact the device pulling either interrupt. The DSP3210 will actually turn off the interrupt it's pulling as part of the DSP to host CPU software interface protocol. The host CPU can in turn generate DSP interrupts by writing lows to either DSP1* or DSP0*. Again, software protocols are necessary to ensure that the host CPU negates either interrupt line at the appropriate time. More information on DSP and host interaction is contained in the DSP3210 chapter.

In normal use, this register should be initialized to $FF before changing any of the data direction control bits to outputs. Failure to do this may cause unhandled interrupts to either host or DSP processors. It may also cause undefined behavior in Coprocessor slot boards that take advantage of the CI2P* and PI2C* lines. An I/O reset will reset the data direction register to all inputs, but has no effect on the stored output values.

2.4.3 SSPB Control

The Synchronous Serial Peripheral Bus (SSPB) interface consists of a data register at $00DD0058 and a control register at $00DD005C. The SSPB bus is a flexible two-wire interface for control of low speed peripherals. It consists of a data line, SP, and a clock line, CNT. The SSPB control registers allow virtually any possible two-wire synchronous serial bus to be implemented under CPU control, but they provide automatic clocking with interrupt-when-done for serial protocols than can be constructed from eleven bits or less of data.

Before any SSPB activity can take place, the SSPB control register (SSPBCTL) must be set up. The bit assignments of the SSPB Control Register is given in *Table 2-8*. The XDONE bit goes high to indicate that a successful transfer has completed; it is cleared on read, a no-op on write. The INTENB bit is written low to disable interrupts, high to cause an interrupt to be generated by the DMAC when a byte has been transferred over the SSPB bus. Interrupts are generally used by all clocked SSPB protocols.

Bit	Name	Function
11	XDONE	Transfer complete
10	INTENB	Interrupt enable
9	ACNT*	Raw access to CNT line
8	ASP*	Raw access to SP line
7-0	COUNT	CNT timebase

Table 2-6: SSPB Control Bit Assignments

A high written to ACNT* will cause a low on the CNT line when the SSPB mechanism is not in a clocked transfer, and a high written to ASP* will cause a low on the SP line when the SSPB mechanism is not in a clocked transfer. ACNT* and ASP* read, respectively, the current state of the CNT and SP lines. This allows the an active programming of the CNT and SP lines, to create serial protocols not supported by the clocked SSPB protocol.

The SSPB automatic clock period is determined by the value written to the CLOCK byte. The cycle time of CNT is given by (CLOCK * 160ns), for values of CLOCK from 1 to 255. The value of CLOCK is preserved over multiple SSPB data cycles.

Once SSPBCTL has been set up, SSPB data transfer cycles may be run. The SSPB data cycle is a little unusual, in that every cycle is technically both a read and write cycle. To start an SSPB cycle, the host processor writes to the 32 bit wide SSPB data register (SSPBDAT) a data value with the lower eleven bits significant. The SSPB mechanism then proceeds to shift out this data, from bit eleven through bit zero. Bit eleven is shifted out before the first clock on CNT, and bit zero is shifted out after the last clock on CNT, as shown in *Figure 2-1*. Once the transfer is complete, the DMAC will bring the system interrupt line low, assuming the INTENB bit is set,

and the host CPU will receive an interrupt, which is held until cleared by reading SSPBCTL. Once the interrupt is cleared, SSPBDAT may be read.

This event sequence holds for either reads from or writes to SSPB devices. For a write, the write data is simply written to SSPBDAT, possibly mixed in with control information, depending on the data format in use. When called in an interrupt server chain, the SSPB server reads SSPBCTL, checking for XDONE. If asserted, SSPBDAT may optionally be checked for any sort of status the protocol in use would demand. For a read, each significant data bit is set high in a write to SSPBDAT. Upon receipt of an interrupt, the interrupt server gets the XDONE bit from SSPBCTL and then the read data and possible status from SSPBDAT.

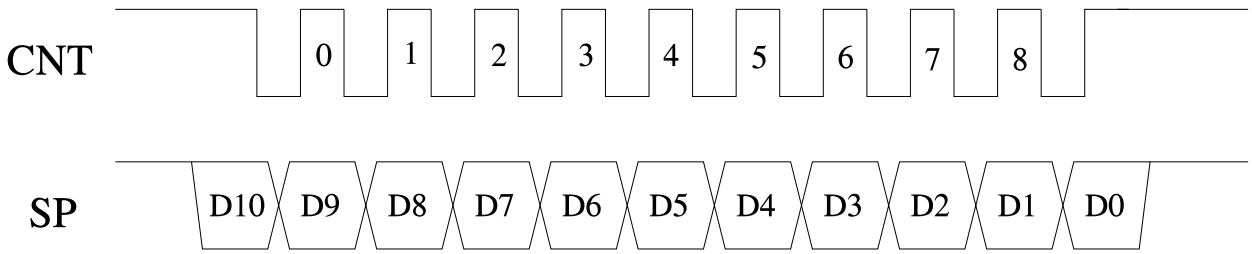


Figure 2-1: SSPB Clock/Data Format

The actual SSPB protocol used depends on what the interface is used for. At present, the A3000+ has two SSPB devices on the motherboard, the battery backed clock/RAM and the audio processor. Both of these devices follow the Phillips I^2C bus protocol. Appendix 4 contains the details on how to configure the SSPB system for I^2C compatible cycles, and also information on the clock and audio processor register maps.

CHAPTER 3

THE PANDORA CHIPS

They're something that I've never seen before

-R.E.M.

The Amiga 3000+ system is supported by a new Amiga gcaphics chip set, called the Pandora, or Advanced Amiga, chip set. This system greatly enhances the Amiga's video display capabilities, while remaining upward compatible with the Amiga ECS chip set, and retaining all of the graphics features unique to previous Amiga systems.

3.1 Basic Description

The Pandora chip set consists of three full custom LSI chips, working together as a single unit. Alice, based directly on the 2MB ECS Agnus chip, is the main Amiga chip bus controller. It is responsible for generating video and memory timing for the other chips, and it contains bimmer and copper units. While Alice is still a 16 bit chip, it now can direct 32 bit and/or double speed transactions on the Chip bus, and is simpler to interface to the A3000 style 32 bit Chip memory bus. Paula is the chip responsible for 8-bit audio output, floppy disk and RS-232 serial I/O, and potentiomenter inputs. The Paula used in the Pandora chip set is the same Paula used in classic and ECS Amiga chip sets.

Completely new for Pandora is the Lisa. Lisa is a new full custom design, a replacement for Denise. Implemented in 1.5μm CMOS, Lisa has a 32 bit Chip bus interface, and with 80ns DRAM, double fetch cycles can fetch 64 bits of data in a single Chip bus cycle, a factor of four improvement over classic and ECS video fetch cycles. The Lisa chip output is 24 bits of digital

RGB video. High quality RBG analog output from the Lisa digital video is generated by an off-the-shelf video DAC, which is currently the Analog Devices ADV7120 Triple 8-bit Video DAC. A summary of Pandora features includes:

- 32 bit wide data bus supports input of 32-bit wide bitplane data and allows doubling of memory bandwidth. Additional doubling of bandwidth can be achieved by using Fast Page Mode RAM. The same bandwidth enhancements are available for sprites. Also the maximum number of bitplanes useable in all modes is increased to eight.
- The Color Palette has been expanded to 256 colors deep and 25 bits wide (8 red, 8 green, 8 blue, 1genlock). This permits display of 256 simultaneous colors in all resolutions. A palette of 16,777,216 colors is available in all resolutions.
- 28Mhz clock input to Lisa allows for cleaner definition of HIRES and SHRES pixels. Alice's clock generator is synchronized by means of Lisa's 14MHz and SCLK outputs.
- A new register bit allows sprites to appear in the screen border regions..
- A bitplane mask field of 8 bits allows an address offset into the color palette. Two 4-bit mask fields do the same for odd and even sprites.
- In Dual Playfield modes, two 4-bitplane playfields are now possible in all resolutions.
- Two extra high-order playfield scroll bits allow seamless scrolling of up to 64 bit wide bitplanes in all resolutions. Resolution of bitplane scroll, display window, and horizontal sprite position has been improved to 35ns in all resolutions.
- A new 8 bitplane HAM mode has been created, 6 bits for colors and 2 for control bits. All HAM modes are available in all resolutions (not just LORES as before).
- A reset input pin has been added, which resets all the bits contained in registers that were new for ECS or LISA.
- Sprite resolution can be set to LORES, HIRES, or SHRES, independent of bitplane resolution.
- Attached Sprites are now available in all resolutions.
- Hardware Scan Doubling support has been added for bitplanes and sprites. This is intended to allow 15KHz screens to be intelligently displayed on a 31KHz monitor, and share the display with 31KHz screens.

While its possible to fully describe each and every display mode, this isn't really necessary. First of all, there are now an awful lot of combinations available. Since the A3000+ supports the full bandwidth Pandora Chip RAM, all display modes are possible. Basically, all display modes, interlacing, and resolution are independent of one another. Display modes can be straight bitmap, from 1 to 8 bitplanes. There is also 6 bitplane Extra HalfBright, and both 6 and 8 bitplane Hold and Modify. Resolutions are a combination of pixel speed and horizontal refresh. Pixels can be 140ns, 70ns, or 35ns. Horizontal refresh rates are usually 15.7kHz or 31kHz. Some of the combinations are given in Appendix 5.

3.2 New Features

There are quite a few new features in the Pandora chip set. The basic details of these features are described here. A more complete explanation is available in the internal Commodore document *Functional Specification for the Advanced Amiga Chip Set (AA)*, by Bob Raible and Spenser

Shanson, where much of this chapter originates. An external version of this document, along with system software changes dealing with the Pandora changes, will certainly be made available as the need for additional information arises.

3.2.1 Bitplanes

There are now 8 bitplanes instead of 6. In single playfield modes they can address 256 colors instead of just 64. As long as the memory architecture supports the bandwidth, all 8 bitplanes are available in all 3 resolutions. In the same vein, 4+4 bitplane dual playfield is available in all 3 resolutions, unless bitplane scan-doubling is enabled, in which case both playfields share the same bitplane modulus register. Bits 15 thru 8 of BPLCON4 comprise an 8 bit mask for the 8 bit bitplane address, XOR'ing the individual bits. This allows the copper to exchange color maps with a single instruction. BPLCON1 now contains an 8 bit scroll value for each of the playfields. Granularity of scroll now extends down to 35nSec.(1 SHRES pixel), and scroll can delay playfield thru 32 bus cycles. Bits BPAGEM and BPL32 in new register FMODE control size of bitplane data in BPL1DAT thru BPL8DAT.

Six Bit HAM

BP_6	BP_5	RED	GREEN	BLUE
0	0	select new base register (1 of 16)		
0	1	hold	hold	modify
1	0	modify	hold	hold
1	1	hold	modify	hold

Eight Bit HAM

BP_2	BP_1	RED	GREEN	BLUE
0	0	select new base register(1 of 64)		
0	1	hold	hold	modify
1	0	modify	hold	hold
1	1	hold	modify	hold

Table 3-1: Pandora HAM Encodings

The old 6 bitplane HAM (Hold And Modify) mode, unlike before, works in HIRES and SHRES resolutions. As before bitplanes 5 and 6 control its function. There is a new 8 bitplane HAM mode. This mode is invoked when BPU field in BPLCON0 is set to 8, and HAMEN is set.
Bitplanes 1 and 2 are used as control bits analagous to the function of bitplanes 5 and 6 in 6 bitplane HAM mode. Since only 6 bitplanes are available for modify data, the data is placed in the 6 MSB. The 2 LSB are left unmodified, which allows creation of all 16,777,216 colors simultaneously, assuming one had a large enough screen and picked one's base registers judiciously. This HAM mode also works in HIRES and SHRES modes. Pandora HAM modes are summerized in *Table 3-1*.

For compatibility reasons EHB mode remains intact. Its existence is rather moot in that we have more than enough colors in the color table to replace its functionality. As before, EHB is invoked whenever SHRES=HIRES=HAMEN=DPF=0,and BPU=6. Please note that starting with ECS DENISE there is a bit in BPLCON2 which disables this mode(KILLEHB).

Bits PF2OF2,1,0 in BPLCON3 determine second playfield's offset into the color table. This is now necessary since playfields in DPF mode can have up to 4 bitplanes. Offset values are as defined in register map.

BSCAN2 bit in FMODE enables bitplane scan-doubling. When V0 bit of DIWSTRT matches V0 of vertical beam counter, BPL1MOD contains the modulus for the display line, else BPL2MOD is used. When scan-doubled both odd and even bitplanes use the same modulus on a given line, whereas in normal mode odd bitplanes used BPL1MOD and even bitplanes used BPL2MOD. As a result Dual Playfield screens will probably not display correctly when scan-doubled.

3.2.2 Sprites

Bits SPAGEM and SPR32 in FMODE determine whether size of sprite load data in SPR0DATA(B) thru SPR7DATA(B) is 16, 32, or 64 bits, analagous to bitplanes. BPLCON3 contains several bits relating to sprite behavior. SPRES1 and SPRES0 control sprite resolution, whether they conform to the ECS standard, or override to LORES, HIRES, or SHRES. BRDRSPRT, when high, allows sprites to be visible in border areas. ESPRM7 thru ESPRM4 allow relocation of the even sprite color map. OSPRM7 thru OSPRM4 allow relocation of the odd sprite color map. In the case of attached sprites OSPRM bits are used.

SSCAN2 bit in FMODE enables sprite scan-doubling. When enabled, individual SH10 bits in SPRxPOS registers control whether or not a given sprite is to be scan-doubled. When V0 bit of SPRxPOS register matches V0 bit of vertical beam counter, the given sprite's DMA is allowed to proceed as before. If they don't match, then sprite DMA is disabled and Lisa reuses the sprite data from the previous line. Sprite vertical start and stop positions must be of the same parity, i.e. both odd or both even.

3.2.3 Video

The BYPASS bit in BPLCON0 allows prioritized color address to bypass the color table and exit Lisa on R(7:0) bus. SOGEN bit in BPLCON2 determines state of Sync-On-Green (SOG) output. This is intended to signal the Video DAC chip (KELLY) to superimpose the composite sync signal onto the analog green output. Horizontal blanking registers HBSTART and HBSTOP have been added to Lisa, eliminating the need for an external blanking input pin. There is now a BLANK output which allows the analog level for blanking to be different than black. BRDRBLNK bit in BPLCON3 allows the background to be BLACK and not the contents of COLOR00 (the name of this bit is now a misnomer, but there are no plans to change it). Note also that the 28MHz output of Lisa provides a pixel-synchronous clock, which will make scan conversion devices much easier to design.

3.2.4 Compatibility

System reset initializes the all new Pandora registers, including BPLCON3, BPLCON4, CLXCON2, DIWHIGH, and FMODE. The ECSENA bit (formerly ENBPLCN3) is used to disable those register bits that are never accessed by old copper lists, and in addition are required by old-style copper lists to be in their default settings. Specifically, a low in ECSENA forces the bits BRDRBLNK, BRDNTRAN, ZDCLKEN, EXTBLKEN, and BRDRSPRT to their default states. When ECSENA is high again, the former settings return. CLXCON2 is reset by a write to CLXCON, so that old game programs will be able to correctly detect collisions. DIWHIGH is reset by writes to DIWSTRT or DIWSTOP. This interlock is inherited from ECS Denise.

3.2.5 Genlock Stuff

Lots of new genlock features were added to ECS Denise and are carried over to Lisa. ZDBPEN in BPLCON2 allows any bitplane, selected by ZDBPSEL2,1,0, to be used as a transparency mask (ZD pin mirrors contents of selected bitplane). ZDCTEN disables the old COLOR00 is transparent mode, and allows the bit31 position of each color in the color table to control transparency. ZDCLKEN generates a 14MHz clock synchronized with the video data that can be used by video post-processors. Finally, BRDNTRAN in BPLCON3 generates an opaque border region which can used to frame live video.

$BANK_2$	$BANK_1$	$BANK_0$	Color Address Range
0	0	0	COLOR00 - COLOR1F
0	0	1	COLOR20 - COLOR3F
0	1	0	COLOR40 - COLOR5F
0	1	1	COLOR60 - COLOR7F
1	0	0	COLOR80 - COLOR9F
1	0	1	COLORA0 - COLORBF
1	1	0	COLORC0 - COLORDF
1	1	1	COLORE0 - COLORFF

Table 3-2: Lisa Color Map Banking

3.2.6 Color Lookup Table

The color table has grown from 32 13-bit registers to 256 25-bit registers. Several new register bits have been added to BPLCON3 to facilitate loading the table with only 32 register addresses. LOCT, selects either the 16 MSB or LSB for loading. Loading the MSB always loads the LSB as well for compatibility, so when 24 bit colors are desired load LSB after MSB. BANK2,1,0 select 1 of 8 32 address banksfor loading, as shown in *Table 3-2*. The RDRAM bit in BPLCON2 causes Lisa to interpret all color table accesses as reads.

There is no longer any need to "scramble" SHRES color table entries. This artifice is no longer required and people who bypass ECS graphics library calls to do their own 28MHz graphics are to be pointed at and publicly humiliated.

3.2.7 Collision

A new register CLXCON2 contains 4 new bits. ENBP7 and ENBP6 are the enable bits for bitplanes 7 and 8,respectively. Similarly, MVBP7 and MVBP8 are their match value bits. CLXDAT is unchanged.

3.2.8 Horizontal Comparators

All programmable comparators with the exception of VHPOSW have 35nS resolution. DIWHIGH, HBSTRT, HBSTOP, SPRCTL, BPLCON1. BPLCON1 has additional high-order bits as well. Note that the horizontal bit position representing 140nS resolution has been changed to 3rd least significant bit, where before it used to be a field's LSB. For example, bit 00 in BPLCON1 used to be named PF1H0 and now it's called PF1H2.

3.2.9 Mouse/Joysticks

A two pin serial mouse interface replaces the 4 pin CCK multiplex scheme. This change is transparent to the programmer and allows 8 extra bits to be shifted in as well. These are displayed in the 8 MSB of LISAID and are intended for hardware configuration information. The mouse counters are updated only 1/16 as often as before (223 kHz), but this should be transparent to the user.

3.2.10 Coercion of 15 kHz to 31 kHz

Lisa contains hardware mechanisms to aid in properly displaying 15 kHz and 31 kHz viewports together on the same 31 kHz display. Lisa can globally set sprite resolution to LORES, HIRES, or SHRES. Lisa will ignore SH10 compare bit in SPRxPOS when scan-doubling, thereby allowing Alice to use these bits to individually set scan-doubling.

CHAPTER 4

THE DSP3210 SUBSYSTEM

A live wire; barely a beginner but just watch that lady go!
-Van Halen

The WE® DSP3210 is a floating-point digital signal processor, extremely well suited for use as a coprocessor in a microcomputer system such as the A3000+. The DSP3210 will speed up many of the things currently done on Amigas, and provide capabilities far beyond those built into current generation Amiga systems.

4.1 WE® DSP3210 Overview

The DSP3210 is a floating-point digital signal processor, capable of running up to four memory accesses per instruction cycle, and up to 12.5 million instructions per second and 25 million floating-point operations per second. The DSP3210 architecture seeks to eliminate the problems that plagued earlier digital signal processors used as graphics, sound, and general signal coprocessors in a microcomputer environment. Rather than being limited to a few hundered bytes of on-chip RAM and a few thousand bytes of off-chip RAM, the DSP3210 couples 8K of very fast on-chip RAM with full 32-bit bus master capability, allowing it access to all 32-bit memory in the A3000+. Additionally, the AT&T Visable Caching Operating System® (VCOS) provides a realtime multitasking kernel for the DSP3210. This allows applications to use the DSP3210 as freely and as easily as they would use the host processor; within the limits of available processor time, no one application can tie up the DSP3210. Also, since VCOS supports multiple DSP3210s, even processor saturation may in time be solved with addtional expansion hardware.

4.2 DSP System Configuration

The basic DSP3210 exists as a normal motherboard bus master, with the highest DMA priority in the A3000+ system. It can run full speed 68030 compatible cycles, including burst-mode to motherboard and Zorro III expansion memory. Since the DSP is not register mapped, the communications protocol between it and the host processor is mainly an agreement in software. The system hardware provides a special DSP reset line which hold the DSP off the bus until the host processor has time to initialize the communications protocol. This reset line is controlled via a register in the Gary+ chip. Once the DSP is up and running, communcations between it and the host processor are helped by the ability to interrupt one another, via the Coprocessor Communication Register, from the host processor side (described in the System Controls chapter), and the DSP3210's BIO port., from the DSP3210 point of view.

The DSP3210 serves two main functions; that of a simple mathematics and signal processing resource, and that of an I/O processor. As a signal processing resource, applications programs can, though the Amiga-VCOS interface, run DSP3210 programs for a variety of jobs. It's also possible that software will be created to use the DSP transparently to the application; a high-level mathematics library is one example of this. Most number processing applications are scheduled under VCOS as non-realtime jobs.

For I/O processing, the DSP3210 has a high speed serial bus which can access the internal RAM via transparent DMA. The A3000+ provides a telepone line CODEC with telephone interface, and a stereo Hi-Fi audio codec with line, microphone, and headphone interfaces. Most I/O interaction, such as realtime audio and modem/FAX applications are scheduled under VCOS as realtime jobs.

4.2.1 DSP Memory Map Differences

While the DSP3210 shares the 32-bit linear address space of the A3000+ host processor, its internal resources are mapped over a small area of A3000+ memory, as shown in *Figure 4-1*. The DSP3210 can't access the memory "beneath" its on-chip resources, and the host processor can't access the DSP3210 on-chip resources. The main on-chip resources are Boot ROM (1K), on-chip I/O (1K), on-chip RAM (8K), and reserved area (54K). Note that the A3000+ system configures the DSP3210 to power up in µComputer mode, causing the on-chip resources to be located from $00000000-$0000FFFF. The alternate µProcessor mode will cause the on-chip resources to be located from $50030000-$5003FFFF.

The other main difference in memory mapping is that the DSP3210 doesn't dynamically size its bus. The A3000+ support logic allows it to communicate with both synchronous and asynchronous 32-bit resources. It cannot, however, access 8 or 16 bit devices, such as some I/O registers, all custom chip control registers, and any Zorro II device. The system hardware is designed to send the DSP3210 a bus error if it attempts access of any non-32 bit wide port. Normally, the DSP is expected to need access only to RAM in any case, and Fast RAM at that most of the time. It should offload any external I/O processing to the host processor.

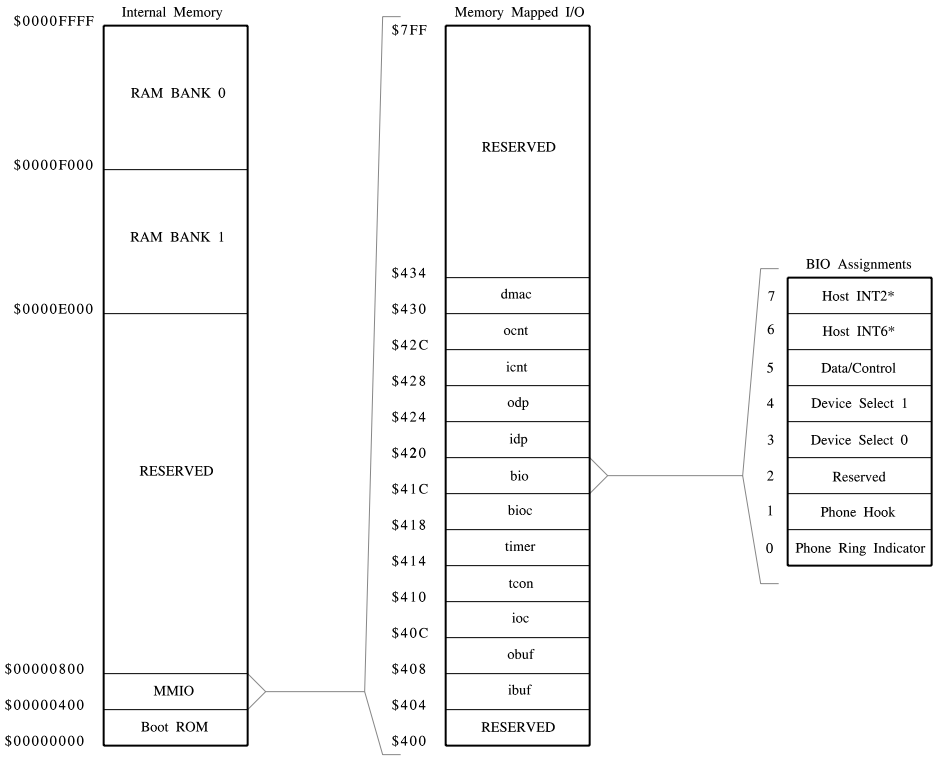


Figure 4-1: DSP3210 Internal Memory Map

4.2.2 The DSP Reset Mechanism

Since the DSP comes up in µComputer mode, the Boot ROM is used for reset. The A3000+ sets things up such that the DSP3210 will expect to execute code starting at $0001000. In order to let the DSP out of reset, the host processor will have to assure that this location contains the necessary DSP boot code. If this memory already contains something, it'll be necessary to first disable interrupts, save off whatever is at $00010000, move the initialization code to $00010000, enable the DSP via the Gary+ DSPRST register, wait on a signal (not an interrupt, these are disabled!) from the DSP indicating its through with the initalization code, then restore $00010000, and finally enable interrupts again. This procedure shouldn't be anywhere nearly as nasty as it sounds, and it only needs to be done once, right after reset.

Typical initialization will go something like this. The DSP wakes up and starts running code at $00010000. This code will more than likely transfer another chunk of code into the DSP's internal memory. The DSP will complete the transfer, jump to the internal code, and then signal the host processor, probably via a memory based semaphore, that it's done with the initialization code. This internal code will, at the least, allow the DSP to accept commands from the host processor. Initially, the DSP will wait for an interrupt. In a real system, interrupts will come from I/O as well as from host processor communications, of course. When the DSP gets a host attention interrupt, it'll go out to an agree-upon command buffer in external memory and figure out just what the host wants. Once it finishes the host task, it may notify the host that the task is complete, again via interrupt.

4.2.3 The BIO Port

The A3000+ system configures the DSP3210 Bit I/O (BIO) port for several system support functions. The port-line relative mapping is shown in *Figure 4-1*. The first two bits are assigned to the telephone line CODEC. The Ring Indicator bit is an input from an optoisolator in the telephone line interface, which indicates to the DSP3210 that the phone is ringing. The next bit, the Hook bit, is an output used to hang up or pick up the phone line. The bit after that, bit two, is currently reserved.

The next three bits are used to configure the serial port device under DSP3210 control. The Device 0 and Device 1 bits select one of four possible high speed serial bus devices. Device zero is the telephone line CODEC, device one is the Hi-Fi Audio CODEC, both located on the motherboard. Devices two and three are, respectively, devices on the DSP expansion port selected, respectively, by EXP0* and EXP1*. The next bit is the data/control mode toggle. Some serial bus devices have a distinction between data mode, which is the normal operating mode, and control mode, which set up various control parameters. The current serial port device powers up as the telephone line CODEC< and is selected by changing the value of the Device 0 and Device 1 bits when in command mode.

The final two bits are part of the communications protocol between the DSP3210 and the host processor. When the DSP needs to interrupt the host processor, it drives the Host INT2* bit low to generate a host bus level 2 interrupt, or the Host INT6* bit low, to generate a host bus level 6 interrupt. The interrupt server routine running on the host needs to signal the DSP3210 to clear these interrupts, otherwise the system will wind up in an infinite interrupt loop.

4.3 DSP Serial I/O

The DSP3210 has a very flexible high speed serial port, as previously mentioned. Under control of the BIO port and external logic, this serial bus manages up to four separate devices, each of which may use its own serial bus protocol. The serial port itself is controlled by the ioc register, show in *Table 4-1*.

The serial bus consists of seven interface lines. One is the frame synchronization line (SY) that's used for both input and output. There are three lines for input: clock (ICK), load strobe (ILD), and data (DI). Similarly, there are hree lines for output: clock (OCK), load strobe (OLD), and data (DO). Via the ioc register, various serial protocols can be set up. Clocks, load strobes, and the frame synchronization line can be set up either as inputs or as outputs, depending on the setting of various bits in the ioc register. The internal versus external settings are all independent of one another. When internal, the timing for ICK and OCK is set by the ICN. The AIC bit determines if ICK is internal or external, while AOC determines if OCK is internal or external. Other internal clocked timing is based on the internal clock, which is either ICK or OCK, as set in the BC field. Internal ILD is based on the internal clock, divided by 32; the AIL bit controls whether ILD is internal or external. Internal OLD can simlarly be based on the internal clock, divided by 32, or it can be run in "burst" mode, based on the status of the output buffer and the DMA controller., as set by the BO bit. Internal frame sync is based on the internal clock, either

Bit	Name	Function	Settings (0,1)
31-24	N/A	Reserved	Read as zero.
32-20	ICN	Internal clock	rate = ICN * 4.
19	OSZ	Ouput DMA size	32 bits, OLEN.
18	ISZ	Input DMA size	32 bits, ILEN.
17	OUT	Output data order	LSB first, MSB first.
16	IN	Input data order	LSB first, MSB first.
15	SAN	Sanity bit control	Sanity clear, sanity set.
14	IIC	ICK polarity	ICK true, ICK inverted.
13	BO	OLD clocked/burst	Clocked, Burst
12-10	OLEN	Ouput data length	8, 16, 32, or 24 bits long.
9	AOL	OLD source	External, internal.
8	AOC	OCK source	External, internal (ICN).
7-6	ILEN	Input data length	32, 8, or 16 bits long.
5	AIL	ILD external/internal	External, internal.
4	AIC	ICK external/internal	External, internal (ICN).
3-2	SLEN	SY frequency	Ratio = 32, 8, 16, or 32.
1	BC	SY source	ICK, OCK.
0	ASY	SY external/internal	External, internal.

Table 4-1: The DSP3210 ioc Register

8, 16, or 32 times the period of the on-chip load signal, based on SLEN. The ASY bit sets frame sync to either internal or external operation.

The serial bus data format is also quite flexible. Both input and output data streams can be set for MSB first or LSB first. The input clock polarity can also be changed. The input data format can be 8, 16, or 32 bits long, initiated by the ILD pulse, or 32 bits long, followed by the ILD pulse, as set by the ILEN field. The output data format can be 8, 16, 32, or 24 bits long, as set by the OLEN register. Input data is read from the ibuf register, output data is written to the obuf register.

4.3.1 The Phone-Line CODEC

The first of the A3000+ serial bus devices is the Phone-Line CODEC. This is a 16-bit bidirectional ΣΔ A/D-D/A converter designed especially for use in telecommunications applications. The device maintains 80 dB S/NR and THD, and supports sample rates of 7.2kHz, 8.0kHz, and 9.6kHz. For advanced transmission protocols, such as V.32, the device contains a digital resampling interpolation filter to all resampling of the receive data at the same phase as the transmitted data. The device is sufficient to support V.33, V.32ter, V.32bis, V.32, V.29, V.27bis, V.27, V.26bis, V.22bis, V.22, Bell 212A, and Bell 103 protocols (at least some of these are supported by VCOS modules available from AT&T).

The Phone-Line CODEC works the same in either control or data modes of the serial bus, and the serial device address is zero. The DSP3210 is set up to use an external clock, with internally generated transmit load strobe, externally generated receive load strobe. The device itself contains nine addressable ixteen-bit registers, five for control purposes, four for data I/O, as show in *Table 4-6*.

All transmissions consist of a 16-bit address word and a 16-bit data word. Writes to the device are initiated by the DSP3210. The address word contains the register address as bits zero through three, zeros for the reserved bits four through fourteen, and a zero as bit fifteen, indicating a write. To read a control register, the DSP3210 sends the address word, only with bit fifteen set high, followed by a dummy data word. This doesn't immediatedly read that register,

Addr	Name	Function
0	CTL0	Rate select, interpolation filter enable
1	CTL1	Rate scaling, power down, filter bypass
2	CTL2	Receive bit and baud rate selects
3	CTL3	Transmit bit and baud rate selects
4	CTL4	Receive phase adjustment
5	CTL5	Transmit phase adjustment
6	DAT0	Transmit data
7	DAT1	Interpolation filter input
8	DAT2	Receive data
9	DAT3	Interpolation filter output

Table 4-2: Phone-Line CODEC Registers

but causes a read request to be logged into the CODEC device. All reads are actually initiated by the device to the DSP3210. It sends a 16-bit address word followed by a 16-bit data word, in the same format as for write transmissions. The address word is used by the DSP3210 to figure out what is being received.

More information on the Phone-Line CODEC, such as the format of the various control registers, is available in Appendix A.6.1. Additional reference material will eventually be made available.

4.3.2 The Hi-Fi Audio CODEC

The second device on the A3000+ DSP serial bus is the Hi-Fi Audio CODEC. This device is a 16-bit bidirectional two-channel ΣΔ A/D-D/A converter. It maintains greater than 80 dB S/NR and THD, and supports a large variety of sample rates, including the audio-standard 44.1kHz and 48kHz rates. It supports programmable data formats, either signed two's complement 16-bit linear, 8-bit μ-law, or 8-bit A-law encoding. Additionally, it has a programmable input source selection for microphone or line-level inputs.

The Audio CODEC, with serial bus device address of one, has distinct control and data modes. Both modes use 64-bit data frames, transmitted LSB first. The control mode frame sets the sampling rate, data format, and various other initialization parameters, while the data mode frame is the normal I/O mechanism for D/A-A/D conversion. In control mode, the DSP3210 supplies both serial clocks, both load strobes, and the frame synchronization pulse. In Data mode, the Audio CODEC supplies both serial clocks and the frame synchronization pulse, while the DSP3210 supplies the load strobes.

The control mode frame sets a number of parameters in the Audio CODEC. It selects the data format; either mono or stereo, 16-bit twos-complement PCM linear, 8-bit μ-law companded, or 8-bit A-law companded. It sets up the sampling configuration, which includes clock source and sampling frequency. It has optional loopback bits, which allow the system to be tested, with either ADC to DAC or DAC to ADC looping. And the revision code of the CODEC is available here too, to help device drivers adjust to future versions of the device.

Data mode, as expected, contains the audio data, as defined in control mode. All frames allow programmable output attenuation on both left and right channels, from 0dB to 93dB, in 1.5dB steps, as well as full digital mute. Analog mute is also available on either output channel. Input gain can be set from 0db to 22.5dB, in increments of 1.5dB. Of course, the input source can be selected as either line or microphone. A programmable monitor mode allows the ADC output to be variably mixed with the DAC's input, in steps of 0dB (fully mixed) to an attenuation of 84dB, in steps of 6dB, as well as completely unmixed. Finally, there's an overrange indicator which indicates if either ADC channel is overdriven.

One important note is the proper protocol must be observed for switching this device between control and data modes. To go from data to control mode, it is only necessary to lower the output volume to its maximum attenuation, then bring the D/C line low, via the bio port. Next, create the desired control frame and send it, with the DCB (Data Control Handshake) bit low. Read back the control frame and examine the DCB bit. Until it reads low, continue sending the control frame and reading it back. This allows multiple Audio CODECs to exist at the same serial bus device address. Once DCB reads back low, the control frame is written back once more, only with DCB set high. Now the CODEC is ready for the new setup. The D/C line is brough high, which will put the device into data mode and also execute an offset calibration on the selected input channel.

More information on the Hi-Fi Audio CODEC, such as the detailed format of the various transmission frames, is available in Appendix A.6.2. Additional reference material will eventually be made available.

4.4 Other DSP Information

The DSP contains an on-chip DMA controller, programmed via the dmac register, counter registers icnt and ocnt, and pointer registers idp and odp. The DMA controller can be used to automate serial bus transfers. There is a timer, which is controlled by the timer register, timer, and the timer control register, tcon. The timer can be used to generate an internal interrupt. The

other MMIO registers, the serial control registers ibuf, obuf, and ioc, plus the BIO port control registers, bio and bioc, have already been discussed.

The DSP3210 itself has quite a few more features. It's a full 32-bit microprocessor in its own right, with 32 bit PC, twenty-two 32-bit general purpose registers, four 40-bit floating-point assumulators, and some control registers. The processor directly supports 16 and 32-bit signed integer and 32-bit floating point numbers, and it can directly load and store 8-bit signed and unsigned and 16-bit unsigned numbers. In addition, it has functions to convert to and from floating point and 8, 16, or 32-bit integer, m-law, A-law, or single-precision IEEE format floating point formats. It also has a great deal of instructions and addressing modes to operate on the internal format data. For more information on the DSP3210, consult WE® DSP3210 Digital Signal Processor, Information Manual, Revision 1.8.1, from AT&T Microelectronics.

CHAPTER 5

SYSTEM EXPANSION

Have you ever wondered, about sound and vision?
-David Bowie

The A3000+ supports the standard forms of expansion found in all slotted Amiga computers; machine-specific Local Bus Slot, for Coprocessor devices and processor upgrades, Amiga standard expansion bus for general purpose expansion, and a Video Slot, for Genlock and other specialized video applications. In addition, the A3000+ motherboard has a few extra expansion headers to provide access to a couple of its special features.

5.1 The Local Bus Slot

The A3000+ Local Bus slot is a superset of the 200 pin Local Bus Slot used in the A3000 and A3000T computers. It is compatible with all A3000 and A3000T coprocessor devices, and adds a few new features. The full pinout of the A3000+ Local Bus Slot is given in Appendix A.2.9, while the slot is fully explained in The A3000 Local Bus Slot Specification, Revision 2.00, which is available from Commodore. The signals new to the A3000+ are as follows:

Slot Interrupt (/INT6)

This is the open-drain, shared, level six interrupt to the host CPU. The coprocessor device can generate this interrupt as long as it supplies the proper interrupt server, and has a way for that server to indentify that it was the interrupt source. Sometimes, one of the Coprocessor communications bits is used for "Interrupt generated" to the host CPU.

Coprocessor Interface (/CI2P, /PI2C)

These lines, as discussed in the Coprocessor Control Register section, are for use by the coprocessor device and the host processor as they see fit for application specific communications. The /CI2P line is earmarked as a Coprocessor Slot to host processor line, the /PI2C line for host processor to Coprocessor Slot signals.

A3000+ Detect (/SENSEA3P)

This line is attached to a 1K pullup resistor on the Coprocessor device board. If it goes high, this is an A3000 system, if low, this is an A3000+system.

5.2 The Zorro III Bus

The Zorro III bus is the standard, general purpse expansion channel on the A3000+. There are no major changes to the Zorro III on the A3000+, though all A3000+ systems are expectecd to ship with the Buster II bus controller, which implements the complete Zorro III specification. The Zorro III Bus is detailed in The Zorro III Bus Specification, Revisin 1.10, which is available separately from Commodore. The Zorro III pinout is given in Appendix A.2.8.

The one new feature supported by the A3000+ is a configurable Zorro II cache support control. Normally, 32 bit Amiga systems map Zorro II memory space geographically into cachable and uncachable. This assumption works for most devices, but can fail for shared memory coprocessors, such as the Commodore-Amiga Bridge Card. Jumper J680 is responsible for switching Zorro II mode to fully uncached.

5.3 The Video Slot

The A3000+ supports an enhanced Video Slot, which is found in-line with the first Zorro III expansion slot. This slot allows expansion access to all digital and analog video signals, clocks, and some I/O lines. This slot is mainly upward compatible with the A2000 and A3000 video slots. The A3000+ Video Slot is physically two one-piece edge card connectors, consisting of one 36 pin edge connector, called the "A" connector, and one 50 pin edge connector, called the "B" connector. The pinouts for these are given in Appendix A.2.7.

5.3.1 Power Connections

The Video Slot provides several different voltages designed to supply Video devices.

Digital Ground

Digital supply ground used by all digital devices in the A3000+.

Audio Ground

Separate ground, decoupled for use by analog audio devices only.

Main Supply (+5V)

Main digital level power supply the Video Slot. This can supply large currents. This connection is speced at 1 Amp, though with proper power budgeting can physically supply as much as 4 Amps.

Negative Supply (-5V)

Negative version of the main supply, for small current loads only; there's a total of 0.3 Amp for the entire A3000+ system.

High Voltage Supply (+12V)

Higher voltage supply, intended for small loading only; there's 500mA reserved for use by the Video Slot.

5.3.2 Clock Signals

These are various clock signals useful for synchronous timing of video peripherals.

/C1 Clock

This is a 3.58 MHz clock (3.55 MHz on PAL systems) that's synched to the falling edge of the 7M system clock. Also known as /CCK in some places.

/C3 Clock

This is a 3.58 MHz clock (3.55 MHz on PAL systems) that's synched to the rising edge of the 7M system clock. Also known as /CCKQ in some places.

/C4 Clock

This is a 3.58 MHz clock (3.55 MHz on PAL systems) that's synched to the rising edge of the 7M CDAC clock.

CDAC Clock

This is a 7.16 MHz clock (7.09 MHz on PAL systems) that trails the 7M system clock by 90°.

C28O Clock

This is a 28.64MHz clock (28.36MHz on PAL systems) that's synchonized to the digital RBG bus.

Timer Time Base (TBASE)

This is the real time clock time-base input, either 50Hz or 60Hz depending on the country involved and the setting of the Time Base Jumper. The jumper can select either line frequency or vertical synchronization as the clock's time base.

External Clock (XCLK, /XCLKEN)

The video slot provides for an external system clock, generally used to cause the entire A2000 system to become synchronized to something external. This should be something very

close to the 28.64 MHz clock normally used to drive the system; the value used for XCLK can be a somewhat higher frequency, though anything too high will cause memory and other system timings to break down. XCLK will only be engaged as the system clock when /XCLKEN is asserted. There is no fixed phase relationship between XCLK and the internal clocks and video output; C28O can always be used for pixel clocking.

5.3.3 Video Signals

The main point of this slot is access to the video signals generated by the Amiga's custom video chips, may of which are not available on the 23 pin external video connector.

Analog Video (Analog Red, Analog Green, Analog Blue)

This is the analog RGB output, which consists of Red, Green, and Blue signals each of which generates a 0.7V p-p, 75 Ohm ternimated, analog output.

Digital Video ($RGB_{23}..RGB_{0}$)

These are the digital video outputs. $RGB_{23}..RBG_{16}$ corresponds to $Red_{7}..Red_{0}$, $RGB_{15}..RGB_{8}$ corresponds to $Green_{7}..Green_{0}$, and $RGB_{7}..RGB_{0}$ corresponds to $Blue_{7}..Blue_{0}$. These are arranged such that the high order four bits of each Red, Green, and Blue correspond to the 12 bits of digital video available in A2000 and A3000 video slots.

Separate Sync (/HSYNC, /VSYNC)

These are the separate, bidirectional, 47 Ohm terminated video frame synchronization clocks, /HSYNC is the horizontal sync, /VSYNC is the vertical sync. These sync signals are typically active low, but can be programmed active high by Alice. Genlock devices source these lines to slave the Amiga video to their input video source.

Composite Sync (/CSYNC)

/CSYNC is an unterminated, buffered, digital level composite sync. The analog composite sync line, COMPSYNC, is not available on the A3000+ video slot.

/BURST

NTSC/PAL colorburst gate, used to remove colorburst during the vertical blanking interval.

Pixel Switch (/PixelSW)

Genlock overlay color indicator, set by a variety of conditions, on a pixel by pixel basis.

SOG

This was originally a programmable sync-on-green line that went to the Kelly A/D converter. Since the demise of Kelly, sync-on-green is done via jumper, and this line is free for use by a video slot device. This line reflects the state of bit seven in BPLCON2.

Light Pen (/LPEN)

This is an input to the Alice light pen input. This signal should go low in response to the lighting of a pixel on a video display monitor. The Alice chip latches the raster position that was in effect when the /LPEN signal goes low, so an application can follow the position of a light pen on the screen.

5.3.4 Audio Signals

Along with access to video signals, audio signals are available at the Video Slot.

Audio Line Out (LineoutLF, LineoutRT)

The LineoutLF and LineoutRT lines are actually duplicated on the A3000+ Video Slot. The original line out function remains intact. However, given the addition of the DSP audio in the A3000+, a raw tap from Paula makes much less sense, so it has been eliminated.

Filter Cutoff (/LED)

This is the /LED port line which, by Amiga convention, is used to cut out the two pole low pass filter on the Paula generated audio channels. When asserted, the filter is in place; when negated the filter is bypassed. This is an input to this Video connector, useful to allow any Audio/Video card to monitor the audio filtering state.

5.3.5 Port Connections

Most of the signals from the bidirectional parallel port (printer port) are available on this connector as well, along with a few others. Note that use of the parallel port lines by a video card may conflict with the external use of this port by a printer or other device.

8 Bit Parallel Port (PD0-PD7)

The 8 bit bidirectional parallel port most commonly used to drive a Centronics interface printer externally is accessable here. It can be used to control various aspects of a complex video interface device.

Parallel Port Handshake (/PACK)

This is the acknowledge input, /PACK, the same as the acknowledge input to the parallel port. Driving this with an output from a Video Card can cause a level 2 interrupt to occur through the 8520 CIA device this is connected to, based on the programming of an 8520 register.

Other Port Lines (PBUSY, PPOUT, PSEL,/PSTROBE)

PBUSY and PPOUT are general purpose I/O signals that together can also function as a synchronous serial data port driven by an 8520 CIA device. In normal printer use, the PBUSY signal is used to indicate printer buffer full to the Amiga, while PPOUT is used to indicate the printer paper is out. For serial port usage, PBUSY is the serial clock, PPOUT is the serial data line. These should be driven with open collector devices if the Video Card uses them as inputs to the 8520. The PSEL signal is a general purpose I/O port line, usually used as a device select

signal on the parallel port. /PSTROBE is a port handshake output, which can be used as a data ready or data accepted strobe by the receiving end of a parallel port transfer.

SSPB Port (SP, CNT)

The A3000+ Synchonous Serial Peripheral Bus is available here, to be used for controlling a wide variety of Audio and Video peripherals compatible with this serial bus. Since the SSPB supports multiple devices, it's a much better port to use than the parallel port for video card control as long as it's fast enough.

5.4 CD/AUX Connector

The CD/AUX connector is an eight pin SIL header, designed to allow a CD-ROM or similar device to be mixed into the A3000+ audio channels. This was originallly conceived for tower machines, which can actually house an internal CD-ROM drive, but it was left in because of its usefulness in the A3000+, even if a CD-ROM drive won't fit internally. The pinout is given in Appendix A.2.2. Basically, it provides +5V, +12V, and ground, enough to power a small chip if necessary. The Audio ground is provided for shielding, and there are left and right channel outputs, which are mixed in with the audio output from the Paula chip. Finally, the SSPB clock and data lines are available here, should a small serial bus device be useful in some kind of interface here.

5.5 DSP Connector

The A3000+ provides a 36 pin header for DSP expansion. The pinout is given in Appendix A.2.3, and additional information on the DSP signals can be found in the DSP chapter and in the DSP3210 specification, WE® DSP3210 Digital Signal Processor, Information Manual, Revision 1.8.1, from AT&T Microelectronics.

5.5.1 Power Connections

The DSP Connector provides several different voltages designed to supply DSP enhancement devices.

Digital Ground

Digital supply ground used by all digital devices in the A3000+. The DSP connector has grounds on every odd pin except the key, to facilitate the use of a short ribbon cable to the connector.

Main Supply (+5V)

Main digital level power supply, which can supply reasonably large currents. DSP enhancement devices must be factored into a full system power budget.

Negative Supply (-5V)

Negative version of the main supply, for small current loads only; there's a total of 0.3 Amp for the entire A3000+ system.

High Voltage Supply (+12V)

Higher voltage supply, intended for small loading only; this must as well be properly budgeted.

Negative High Voltage Supply (-12V)

Negative version of the high voltage supply, for small current loads only; there's a total of 0.3 Amp for the entire A3000+ system.

5.5.2 DSP Serial Bus

The raw DSP3210 serial bus is brought out to the DSP connector. A variety of serial protocols can be implemented with these lines and a PAL or so.

Frame Synchronization (SY)

This line, which can be generated by the DSP3210 or an external source, is used to synchronize frames, for serial formats that require multiple word packet frames. Internally generated frame syncs can come from either ICK or OCK.

Serial Input Port (DI, ICK, /ILD)

The DI line is the serial input. ICK is a clock driving this port, and it can sourced by the DSP3210 or an external device. /ILD is the load strobe, which can signal the start or finish of a word transfer, and can be sourced internally or externally.

Serial Output Port (DO, OCK, /OLD)

The DO line is the serial output. OCK is a clock driving this port, and it can be sourced by the DSP3210 or an external device. /OLD is the load strobe, which can signal the start or finish of a word transfer, and can be sourced internally or externally.

5.5.3 Other Signals

There is a mix of other useful signals on the DSP expansion connector.

DSP Reset (/RST)

This is the reset line which starts up the DSP.

CODEC Select (/EXP1, /EXP0)

The A3000+ logic allows one of four devices to run on the serial bus at any given time. Two are on-board, the phone line CODEC and the audio CODEC. The /EXP1 and /EXP0 lines are selects for two more, which could exist on some DSP expansion device. These lines are controlled by DSP port lines, as explained in the DSP3210 chapter.

Data/Control (D/C)

This line goes high for data mode transfers, low for control mode transfers. Not every serial bus device will have distinct data and control modes, but some do. In any case, the device

running the serial bus can only change during control mode, where it is generally assumed that all devices will listen for their device selects and get off the serial bus if they are not selected.

SSPB Port (CNT, SP)

The SSPB serial port is made available here, should any low speed peripherals be useful in the design of a DSP enhancement device.

APPENDICES

A.1 External Connector Pinouts

The A3000+ has a considerable number of external ports. While not all of them are new to the A3000+, they are all covered here anyway, just for completeness.

A.1.1 Mouse Port

Pin Number		Mouse	Joystick	Other
1	V Pulse	Forward		
2	H Pulse	Back		
3	VQ Pulse	Left		
4	VH Pulse	Right		
5	Middle Button	Button 3	Analog Pot X	
6	Left Button	Button 1		Light Pen Beam
7	+5V	+5V	+5V	
8	Ground	Ground	Ground	
9	Right Button	Button 2	Analog Pot Y	

A.1.2 Keybord Port

Pin Number	Signal Name
1	No Connection
2	Data
3	Clock
4	+5V
5	Ground

A.1.3 Serial Port

Pin Number	Signal Name
1	Ground
2	TxD
3	RxD
4	RTS
5	CTS
6	DSR
7	Ground
8	CD
9	+12V
10	-12V
11	Audio Out
12-17	No Connection
18	Audio In
19	No Connection
20	DTR
21	No Connection
22	RI
23-25	No Connection

A.1.4 Parallel Port

Pin Number	Signal Name
1	/STROBE
2	Data 0
3	Data 1
4	Data 2
5	Data 3
6	Data 4
7	Data 5
8	Data 6
9	Data 7
10	/ACK
11	BUSY
12	POUT
13	SEL
14	/AUTO (+5V)
15	No Connection
16	/INIT
17-25	Ground

A.1.5 Floppy Port

Pin Number	Signal Name
1	/RDY
2	/DKRD
3-7	Ground
8	/MTRX
9	/SEL3
10	/DKRST
11	/CHNG
12	+5V
13	SIDE
14	/WPROT
15	/TRK0
16	/DKWEB
17	/DKWDB
18	STEP
19	DIR
20	No Connection
21	/SEL2
22	/INDEX
23	+12V

A.1.6 SCSI Port

Pin Number	Signal Name
1	/REQ
2	MSG
3	/ID
4	/RESET
5	/ACK
6	/BSY
7	Ground
8	/Data 0

9	Ground	9	Digital Red
10	/Data 3	10	/CSYNC
11	/Data 5	11	/HSYNC
12	/Data 6	12	/VSYNC
13	/Data 7	13	Ground
14	Ground	14	Pixel Switch
15	/CD	15	/C1
16	Ground	16-20	Video Ground
17	/ATN	21	-5V
18	Ground	22	+12V
19	/SEL	23	+5V
20	/Parity		
21	/Data 1		
22	/Data 2		
23	/Data 4		
24	Ground		
25	Terminal Power		

A.1.7 Audio Port

Pin Number	Signal Name
1	Line In
2	Audio Ground
3	Mic In -
4	Audio Ground
5	Mic In +
6-11	Audio Ground
12	Line Out
13	+5V
14	Alternate Out
15	Audio Ground

A.1.8 Video Port

Pin Number	Signal Name
1	XCLK
2	/XCLKEN
3	Analog Red
4	Analog Green
5	Analog Blue
6	Digital Intensity
7	Digital Blue
8	Digital Green

A.2 Internal Connector Pinouts

The A3000+ has a considerable number of external ports. While not all of them are new to the A3000+, or even generally available to the user, they are all covered here anyway for completeness.

A.2.1 Power Connector

Pin Number	Signal Name
1	Video +5V
2-5	Main +5V
6-10	Ground
11	-5V
12	+5V User
13	TICK
14	-12V
15	+12V

A.2.2 CD/AUX Connector

Pin Number	Signal Name
1	Mix in left
2	Audio Ground
3	Mix in right
4	+12V
5	+5V
6	CNT (SSPB Clock)
7	SP (SSPB Data)
8	Ground

A.2.3 DSP Connector

Pin Number	Signal Name
1,5-35 odd	Ground
2,4	+5V
6	-5V
8	+12V
10	-12V
12	SY
14	DI
16	ICK
18	/ILD
20	DO
22	OCK
24	/OLD
26	/RST
28	/EXP1
30	/EXP0
32	D/C
34	CNT (SSPB Clock)
36	SP (SSPB Data)

A.2.4 Floppy Connector

Pin Number	Signal Name
1,5-33 odd	Ground
2	/CHNG
3	KEY
4	/INUSE1
6	/INUSE0
8	INDEX
10	/SEL0
12	/SEL1
14	/INUSE1
16	/MTROD
18	DIR
20	/STEP
22	/DKWDB
24	/DKWEB
26	/TRK0
28	/WPROT
30	/DKRD
32	/SIDE
34	/RDY

A.2.5 SCSI Connector

Pin Number	Signal Name
1-17 odd	Ground
2	/Data 0
4	/Data 1
6	/Data 2
8	/Data 3
10	/Data 4
12	/Data 5
14	/Data 6
16	/Data 7
18	/Parity
19-24	Ground
25	KEY
26	Terminal Power
27-30	Ground
31-49 odd	Ground
32	/ATN
34	No Connection
36	/BSY
38	/ACK
40	/RESET
42	MSG
44	/SEL
46	/CD
48	/REQ
50	/ID

A.2.6 Daughterboard Slot

The front daughterboard slot uses the Zorro III pinout. This is the back slot, which contains signals for the extra Zorro III slots and the Video Slot on the A3000+ daughterboard.

Pin Number	Signal Name
1	Audio Line Out Left
2	Audio Line Out Right
3	C28O
4,6	+5V
5	Analog Red
7	Ground
8	+12V
9	Analog Green
10,11	Ground
12	/CSYNC
13	Analog Blue
14	/XCLKEN
15	Ground
16	/BURST
17	No Connection
18,19	Ground
20	/HSYNC
21	RGB_4
22	Ground
23	RBG_7
24	/VSYNC
25	RGB_{15}
26	No Connection
27	RGB_{23}
28	/Pixel Switch
29	-5V
30	Ground
31	XCLK
32	/C1
33	+5V
34	PSTROBE
35-37	+12V
38-40	+5V
41	Ground
42	RGB_{20}
43	RGB_{21}
44	RGB_{22}
45	Ground
46	RGB_{12}
47	RGB_{13}
48	RGB_{14}
49	Ground
50	RGB_5
51	RGB_6
52	Ground
53	SOG
54	TBASE
55	CDAC
56	PPOUT
57	/C3

58	PBUSY
59	/LPEN
60	/PACK
61	PSEL
62	Ground
63	PPD_0
64	PPD_1
65	PPD_2
66	PPD_3
67	PPD_4
68	PPD_5
69	PPD_6
70	PPD_7
71	/LED
72	Ground
73	SP (SSPB Data)
74,76	Audio Ground
75	CNT (SSPB Clock)
77	Ground
78	$/EBR_3$
79	$/EBR_2$
80	$/EBR_1$
81	$/EBG_3$
82	$/EBG_2$
83	$/EBG_1$
84	$/SLAVE_3$
85	$/SLAVE_2$
86	$/SLAVE_1$
87	RGB_{16}
88	RGB_{17}
89	RGB_{18}
90	RGB_{19}
91	RGB_8
92	RGB_9
93	RGB_{10}
94	RGB_{11}
95	RGB_0
96	RGB_1
97	RGB_2
98	RGB_3
99,100	Ground

A.2.7 Video Slot

As on the A3000, the A3000+ Video Slot is a two piece edge connector, upward compatible with A2000 and A3000 Video Slots.

A.2.7.1 Video Slot A

Pin Number	Signal Name
1	SP (SSPB Data)
2	CNT (SSPB Clock)
3	Audio Line Out Left
4	Audio Line Out Right
5	C28O
6,8	+5V
7	Analog Red
9	Ground
10	+12V
11	Analog Green
12,13	Ground
14	/CSYNC
15	Analog Blue
16	/XCLKEN
17	Ground
18	/BURST
19	/C4
20,21	Ground
22	/HSYNC
23	RGB_4
24	Ground
25	RBG_7
26	/VSYNC
27	RGB_{15}
28	No Connection
29	RGB_{23}
30	/Pixel Switch
31	-5V
32	Ground
33	XCLK
34	/C1
35	+5V
36	PSTROBE

A.2.7.2 Video Slot B

Pin Number	Signal Name
1	Ground
2	RGB_{20}
3	RGB_{21}
4	RGB_{22}
5	Ground
6	RGB_{12}
7	RGB_{13}
8	RGB_{14}
9	Ground
10	RGB_{5}
11	RGB_{6}
12	Ground
13	SOG
14	TBASE
15	CDAC
16	PPOUT
17	/C3
18	PBUSY
19	/LPEN
20	/PACK
21	PSEL
22	Ground
23	PPD_{0}
24	PPD_{1}
25	PPD_{2}
26	PPD_{3}
27	PPD_{4}
28	PPD_{5}
29	PPD_{6}
30	PPD_{7}
31	/LED
32	Ground
33	Audio Line Out Left
34,36	Audio Ground
35	Audio Line Out Right
37	RGB_{16}
38	RGB_{17}
39	RGB_{18}
40	RGB_{19}
41	RGB_{8}
42	RGB_{9}
43	RGB_{10}
44	RGB_{11}
45	RGB_{0}
46	RGB_{1}
47	RGB_{2}
48	RGB_{3}
49,50	Ground

A.2.8 Zorro III Bus Slot

Pin Number	Signal Name
1-4	Ground
5,6	+5VDC
7	/OWN
8	-5VDC
9	$/SLAVE_{N}$
10	+12VDC
11	$/CFGOUT_{N}$
12	$/CFGIN_{N}$
13	Ground
14	/C3
15	CDAC
16	/C1
17	/CINH
18	/MTCR
19	$/INT_{2}$
20	-12VDC
21	A_{5}
22	$/INT_{6}$
23	A_{6}
24	A_{4}
25	Ground
26	A_{3}
27	A_{2}
28	A_{7}
29	/LOCK
30	AD_{8}
31	FC_{0}
32	AD_{9}
33	FC_{1}
34	AD_{10}
35	FC_{2}
36	AD_{11}
37	Ground

38	AD_{12}	83	SD_4
39	AD_{13}	84	SD_6
40	Reserved	85	Ground
41	AD_{14}	86	SD_5
42	Reserved	87-90	Ground
43	AD_{15}	91	$SenseZ_3$
44	Reserved	92	7M
45	AD_{16}	93	DOE
46	/BERR	94	/IORST
47	AD_{17}	95	/BCLR
48	/MTACK	96	Reserved
49	Ground	97	/FCS
50	E Clock	98	$/DS_1$
51	$/DS_0$	99,100	Ground
52	AD_{18}		
53	/RESET	**A.2.9 Local Bus/Coprocessor Slot**	
54	AD_{19}		
55	/HLT	Pin Number	Signal Name
56	AD_{20}		
57	AD_{22}	1	$/DSACK_1$
58	AD_{21}	2,3	Ground
59	AD_{23}	4	/HALT
60	$/BR_N$	5	R/W
61	Ground	6,7	Ground
62	/BGACK	8	/BGACK
63	AD_{31}	9	/SBR
64	$/BG_N$	10,11	Ground
65	AD_{30}	12	/AVEC
66	/DTACK	13	EXT90
67	AD_{29}	14,15	+5V
68	READ	16	/RAMSLOT
69	AD_{28}	17	/BOSS
70	$/DS_2$	18,19	+5V
71	AD_{27}	20	FC_0
72	$/DS_3$	21	/STERM
73	Ground	22,23	+5V
74	/CCS	24	FC_1
75	SD_0	25	/BR
76	AD_{26}	26,27	+5V
77	SD_1	28	/CBACK
78	AD_{25}	29	/BERR
79	SD_2	30	DIS_CLKS
80	AD_{24}	31	/EMUL
81	SD_3	32	/CBREQ
82	SD_7	33	A_8

34	ECLK30	83	Ground
35	Ground	84	A_{28}
36	A_0	85	A_{21}
37	A_9	86,87	Ground
38,39	Ground	88	A_{29}
40	A_1	89	A_{22}
41	A_{10}	90	Reserved
42	$ECLK90_A$	91	$/DSACK_0$
43	/INT6	92	A_{30}
44	A_2	93	A_{23}
45	A_{11}	94,95	+5V
46	ECLK90	96	A_{31}
47	Ground	97	/DS
48	A_3	98,99	+5V
49	A_{12}	100	/ECS
50,51	Ground	101	/CIOUT
52	A_4	102,103	+5V
53	A_{13}	104	/DBEN
54	$ECPUCLK_B$	105	/BG
55	/WAIT	106,107	+5V
56	A_5	108	/RMC
57	A_{14}	109	/CPURST
58	$ECPUCLK_A$	110	/FPURST
59	Ground	111	Reserved
60	A_6	112	EXTCPU
61	A_{15}	113	/EBCLR
62,63	Ground	114	/SENSEA3P
64	A_7	115	Ground
65	A_{16}	116	/IPEND
66	Reserved	117	/RESET
67	/CI2P	118,119	Ground
68	A_{24}	120	$/IPL_0$
69	A_{17}	121	SIZ_0
70	Reserved	122,123	Ground
71	Ground	124	$/IPL_1$
72	A_{25}	125	FC_2
73	A_{18}	126	CLK90
74,75	Ground	127	Reserved
76	A_{26}	128	$/IPL_2$
77	A_{19}	129	SIZ_1
78	DIS_CLK30	130,131	Ground
79	/PI2C	132	/CIIN
80	A_{27}	133	/AS
81	A_{20}	134	/FPUCS
82	Reserved	135	$CPUCLK_A$

136	/OCS	190,191	Ground
137	D_{31}	192	D_5
138,139	Ground	193	D_{22}
140	D_{15}	194,195	Ground
141	D_{30}	196	D_6
142,143	Ground	197	D_{23}
144	D_{14}	198,199	Ground
145	D_{29}	200	D_7
146	Reserved		
147	/CBR		
148	D_{13}		
149	D_{28}		
150	Reserved		
151	Ground		
152	D_{12}		
153	D_{27}		
154,155	Ground		
156	D_{11}		
157	D_{26}		
158	Reserved		
159	/BG30		
160	D_{10}		
161	D_{25}		
162	Reserved		
163	Ground		
164	D_9		
165	D_{24}		
166,167	Ground		
168	D_8		
169	D_{16}		
170,171	Reserved		
172	D_0		
173	D_{17}		
174,175	+5V		
176	D_1		
177	D_{18}		
178,179	+5V		
180	D_2		
181	D_{19}		
182,183	+5V		
184	D_3		
185	D_{20}		
186,187	+5V		
188	D_4		
189	D_{21}		

A.3 System Configuration Jumpers

There are various strip-post jumpers (links) on the A3000 family motherboards, which control clock speed, sourcing, and other local slot related features.

J100 CLK90 Delay Jumper

This jumper has two positions. In position 1-2, it sets up CLK90 for 25MHz operation. In position 2-3, the EXT90 line drives CLK90, rather than the on-board clock logic.

J102 Board Clock

This jumper has two positions. In position 1-2, the source for CPUCLK$_A$ and CPUCLK$_B$ must be EXTCPU. In position 2-3, the 68030 and local bus clocks all derive from the same source, which is either EXTCPU or the on-board clock, depending on J104.

J103 FPU Chip Select Jumper

This jumper has two positions. With the shunt on pins 1-2, it enables the on-board FPU. With the shunt in position 2-3, disables the on-board FPU.

J104 CPU Clock

This jumper has two positions. In position 1-2, the source for CLK30 is derived from the on-board clock generator. In position 2-3, the source for CLK30 must come from EXTCLK by way of J102.

J105 System Clock Disable

In the 2-3 position, this jumper allows the DIS_CLKS line from the Local Bus Slot to operate normally. In the 1-2 position, it forces DIS_CLKS high (asserted), disabling on-board system clock generation.

J107 68030 Clock Disable

In the 2-3 position, this jumper allows the DIS_CLK30 line from the Local Bus Slot to operate normally. In the 1-2 position, it forces DIS_CLK30 high (asserted), disabling on-board 68030 clock generation.

J120 DSP Burst Select

In the 1-2 position, the DSP will not emulate 68030 burst cycles during block moves. In the 2-3 position, the DSP interface logic will run 2-cycle 68030 bursts in response to quad word block move cycles by the DSP.

J180 Lisa Configuration Options

This is a block of six jumpers that are read by Lisa via the second half of the mouse port register. None of these six jumpers currently have function assignments.

J191 Clock Type

Selects the clock type: position 1-2 for the Phillips, position 2-3 for the SGS-Thomson.

J200 NTSC/PAL Select

Selects NTSC or PAL power up for Alice.

J201 ChipRAM Size

Position 1-2 selects 2M decoding for Chip RAM, position 2-3 selects 8M decoding. The current Alice only supports 2M decoding.

J202 ROM Speed

This selects proper ROM speed. Position 1-2 sets timing for 280ns ROMs, position 2-3 sets timing for 160ns ROMs.

J350 Time Base

This selects the time base souce for the run-time clock in CIA1. Position 1-2 uses the 50/60Hz power supply tick, position 2-3 uses the vertical sync line.

J351 DF1: Recognition Code

Position 1-2 enables the automatic generation of a floppy ID code for 3.5" 880K floppy at unit 1. Position 2-3 disables ID generation by the motherboard for the unit 1 floppy drive.

J450 Sync-On-Green

Position 1-2 sets the system for standard separate sync monitors, position 2-3 sets the system for monitors that want composite sync on the analog green line.

J680 Zorro II Cache

Position 1-2 enables caching for Zorro II memory space ($00200000-$009fffff), position 2-3 disables caching for Zorro II memory space.

J852 Fast RAM Size

In the 1-2 position, RAMSEY generates control for 4MB DRAM, in the 2-3 position RAMSEY generates control for 1MB DRAM.

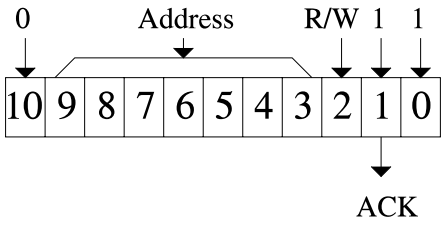
A.4 I²C Bus Information

The I²C Bus is a two-wire serial bus invented by Phillips, for simple connection between multiple low-speed integrated circuits. The A3000+ provides a flexible synchronous serial bus that can, when correctly configured, run I²C bus transactions. The A3000+'s SSPB bus is configured for I²C protocols for communication to the on-board time of day clock and audio processor. Additionally, it may be used in I²C or other SSPB configurations via both DSP connector, auxiliary audio connector, and Enhanced Video Slot. For detailed information on the I2C bus, refer to "The I²C Bus Specification", from Phillips Components.

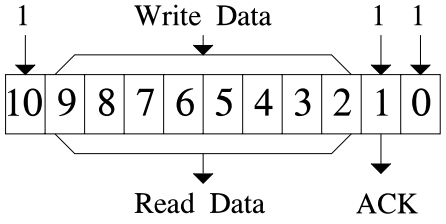
A.4.1 Configuring SSPB for I²C protocols

The main protocol driver for I²C on SSPB must be software, as for most SSPB protocols. Since I²C defines a clock cycle not shorter than 9.4µS, for symmetric clocks, the SSPB clock register must use a value of $3B or greater to generate correct I2C clock timing. Since this will be a clocked transaction, ACNT* and ASP* should be written low, and INTENB should be high. Thus, the SSPB control register is set to $0000043B.

The I²C bus has several basic data formats. The first thing a bus master (eg, the host processor, by way of the DMAC's SSPB ports) does is send out an address. I²C data cycles are always eight bits long; the address cycle starts with a 7 bit address, then a bit indicating whether the next bytes will be reads or writes. Ahead of the address, though, the master needs to present a START condition, which is a high-to-low transition on the data line before any clocks take place. To pack an I²C start, address, and R/W into an SSPB data word, a format is created as illustrated here. The leading bit, bit ten, is written low, to create an I²C start condition. The address immediately follows, then the R/W bit. The last two bits are written high. When the SSPB word has been sent, read back the data register and examine bit one, which should have been driven low, an acknowledge, if an I²C device responded to the address sent. If no device is present at that address, the acknowledge bit will return high. This is important, because it allow software to determine if a particular bus address has a device on it or not.

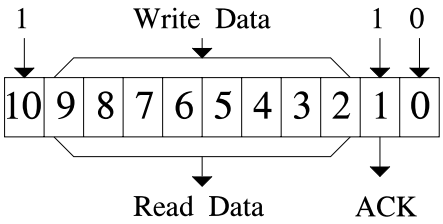


Should a particular device be found, additional I²C bus cycles will send or receive data. The I²C protocol only defines that the data is any number of bytes, one sent or received per bus cycle. The bus master's software should know what kind of device is at any given address, and what that device expects to send or receive in any given transaction. In any case, as mentioned, the I²C protocol defines a stream of bytes that are transferred based on the bus clock. This can be encoded into the SSPB data format as shown here. Bit ten is kept high this time, with bits nine through two set to the data byte, for a write, or $FF, for a read. Bits one and zero are set high. When the



SSPB word has been sent, bit one of the data register will again contain an acknowledge bit, set low, if the transfer was successful. If this is a read, bits nine through two will also contain the read data byte.

Any number of bytes can be sent in this fashion, based only on the requirements of the device being accessed. Eventually, though, the transfer is going to be complete. The I^2C bus defines another special state, the STOP condition, to terminate this data transfer. In the SSPB data word, a normal I^2C data transfer is set up, only in this case, bit zero is set low instead of high. This will cause SP to transition from low to high after CNT is high, ending the transmission. Note also that a STOP isn't necessary if an immediate START will be following it; the START condition forces an implicit STOP. As usual, the acknowledge bit and read data, if any, can be read from the SSPB data word after transmission. Note that it is typically unnecessary to require any changes to the SSPB control word as long as one protocol type is being used on SSPB, and that protocol type fits within the constraints of the clocked transmission model, as does the I^2C bus.



A.4.2 A3000+ Audio Processor

The A3000+ uses an I2C bus device, the Phillips TDA8420/TDA8421 Hi-Fi Stereo Audio Processor, to help manage some of the system audio resources. This device has two stereo input channels and two stereo output channels. It has volume and stereo/mono control over both output channels, as well as bass, treble, and special effects on channel one. In the A3000+, audio directly from the Paula chip, the traditional Amiga audio, goes to channel one. On channel two is an even mix of audio from Paula and from the Audio CODEC. The output channel one goes to the headphone output on the A3000+, while output channel two is the line level output available on the audio outputs and the Enhanced Video Slot.

The device is located at bus address $40, and has eight internal eight-bit wide registers. The second byte sent to this device after the I^2C bus address is a secondary address byte indicating which register is being accessed. Following that byte, the actual data for the register is given. This table indicates the registers and the bit fields within them:

Addr	Name	Function	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	VL1	CH1 Volume Left	1	1	V05	V04	V03	V02	V01	V00
1	VR1	CH1 Volume Right	1	1	V15	V14	V13	V12	V11	V10
2	BA	CH1 Bass	1	1	1	1	BA3	BA2	BA1	BA0
3	TR	CH1 Treble	1	1	1	1	TR3	TR2	TR1	TR0
8	S1	Switch	1	1	MU	EFL	STL	ML1	ML0	IS
4	VL2	CH2 Volume Left	1	1	V25	V24	V23	V22	V21	V20
5	VR2	CH2 Volume Right	1	1	V35	V34	V33	V32	V31	V30
C	S2	Switch	1	1	1	1	EXS	MH1	MH0	1

The Vx5-Vx0 bits control volume for their respective channels. Volume on channel one varies

from -90dB, at settings of $17 or less, to 16db at $FF, in steps of 2dB. Volume on channel two varies from -90dB at settings of $1F or less, to 0dB at $FF, in steps of 2dB. Bass control, for channel one only, varies from -12dB at $2 or less to 15dB at $B or more. Treble, also for channel one only, varies from -12dB at $2 or less to 12dB at $A or more. Both bass and treble adjustments are in steps of 3dB.

There are two switch fields. The active high Mx0 bit selects the left input for transmission to the output channel, while the active high Mx1 select the righ input for transmission to the output channel. When both are selected, stereo output will result from stereo input. The first switch field also contains the MU bit, which mutes all output when set to one (also the powerup default), passes output normally when set to zero. Also in that byte are the STL and EFL bits, which set up the special effects for channel one. Normal stereo is selected with STL high, EFL low. The "spatial" effect is selected with STL and EFL high. A pseudo-stereo effect is generated from a monoural input with the STL bit low and EFL bit high. The final bit in the first switch field is the IN bit, which selects the input source for both outputs. If IN is low, input channel one is selected, if high, input channel two is selected. Finally, switch byte two contains the EXS bit. When EXS is brought low, the external switch pin on the device is brought high impedence, when high, the external switch pin is brought to ground. The A3000+ does not use this external switch pin, but future systems may.

That is the basic description of the sound processor device. More information on this is available in the TDA8420/TDA8421 specifications from Phillips Components.

A.4.3 A3000+ RAM/Clock

The A3000+, like the A3000, supports a battery backed clock with a small amount of RAM in it. Unlike the A3000, the A3000+ uses an I^2C based RAM Clock. At the time of this writing, the final selection of the clock has not been made. The A3000+ motherboard supports both the Phillips PCF8583 clock and the SGS-Thomson MK41T56, selected via jumper J191. While the SGS part appears to be the more desirable of the two, the Phillips part appears to be the more available of the two. The clocks will appear at different I^2C bus addresses, so software can easily determine which clock is in any given A3000+ system.

A.4.3.1 The Phillips PCF8583

This device contains a low power 256x8 bit CMOS RAM and a clock with calendar function. The first 8 bytes of RAM are used for clock functions, the next 8 bytes can be used as RAM or with the alarm clock function. The clock sits at bus address $51. To access a register, the I^2C write cycle is run, with the I^2C address followed by a clock register number. The value to be written to that register follows. Successive bytes will written to successive locations in the clock RAM, thanks to an automatic address increment feature. The same can be done for reads. An I^2C write cycle is done initially, with slave address, write condition, clock RAM address, and then a new start condition. That start is followed by an I^2C read cycle. Every byte after the I^2C address will be read from the clock, using the aforementioned auto-address increment feature.
The register map is given by the following table. Note that the alarm functions really aren’t used

for anything in the A3000+ at present.

Addr	Function	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	Control/Status	STOP	HOLD	----FUNC----		MASK	ALEN	ALF	TIF
1	1/100 Seconds	----------1/10th----------				----------1/100th----------			
2	Seconds	----------10s----------				----------1s----------			
3	Minutes	----------10m----------				----------1m----------			
4	Hours	12/24	PM/AM	-----10h-----		----------1h----------			
5	Year/Date	-----YEAR-----		-----10d-----		----------1d----------			
6	Weekday/Month	------WEEKDAY------		10M		----------1M----------			
7	Timer	----------10d----------				----------1d----------			
8	Alarm Control	ALINT	TAEN	----ALFN-----		TIEN	---------TIFN--------		
9	Alarm 1/100 Seconds	----------1/10th----------				----------1/100th----------			
A	Alarm Seconds	----------10s----------				----------1s----------			
B	Alarm Minutes	----------10m----------				----------1m----------			
C	Alarm Hours	12/24	PM/AM	-----10h-----		----------1h----------			
D	Alarm Year/Date	-----YEAR-----		-----10d-----		----------1d----------			
E	Alarm Weekday/Month	------WEEKDAY------		10M		----------1M----------			
F	Alarm Timer	----------10d----------				----------1d----------			
10-FF	Free RAM								

The basic control register is at location zero. The STOP bit is set low for counting, high for reset. The HOLD bit is set low for founting, high to freeze the count. The FUNC field determine which function the part performs; the A3000+ sets this to 00, for standard 32.768kHz based clock. The MASK value, when high, lets the date and month count be read with year and weekday bits cleared. When MASK is low, year and weekday bits are also read. The ALEN bit is set low to disable the alarm control register, high to enable it. The ALF bit is the alarm flag, the TIF bit is the timer flag.

Most time encodings are straight binary-coded decimal. When the 12/24 bit is high, AM/PM counting is used, when low, 24 hour counting is used. The PM/AM flag goes high for PM, low for AM. The 10h field counts hours from 0 to 2. The year field counts years from 0 to 3, and the 10d field counts days 0 to 3.

Any more information on this part can be found in the Phillips data sheet, "PCF8583: Clock Calendar with 256x8-Bit Static RAM", available from Phillips Components.

A.4.3.2 The SGS-Thomson MK41T56

his device contains a low power 64x8 bit CMOS RAM and a clock with calendar function. The first 8 bytes of RAM are used for clock functions.. The clock sits at bus address **$??**. To access a register, the I^2C write cycle is run, with the I^2C address followed by a clock register number. The value to be written to that register follows. Successive bytes will written to successive locations in the clock RAM, thanks to an automatic address increment feature. The same can be

done for reads. An I^2C write cycle is done initially, with slave address, write condition, clock RAM address, and then a new start condition. That start is followed by an I^2C read cycle. Every byte after the I^2C address will be read from the clock, using the aforementioned auto-address increment feature. The register map is given by the following table.

Addr	Function	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	Seconds	STOP	----------10s----------			----------1s----------			
1	Minutes	X	----------10m----------			----------1m----------			
2	Hours	X	X	-----10h-----		----------1h----------			
3	Weekday	X	X	X	X	X	-----WEEKDAY----		
4	Date	X	X	-----10d-----		----------1d----------			
5	Month	X	X	X	10M	----------1M----------			
6	Year	----------10y----------				----------1y----------			
7	Control	OUT	FT	S	----------CALIBRATION----------				
8-63	Free RAM								

All the time fields are binary coded decimal, and pretty self-explanatory. When the STOP bit set brough high, it causes the clock to stop counting. Bringing it low again resumes the count. The OUT bit is the output of the final counter stage, which runs at 512Hz. When FT bit is high, the OUT signal is driven on the FT/OUT pin of the device, providing a measurable calibration point. When FT is low, the FT/OUT pin instead reflects the value of bit seven of register three.

The S bit is the sign for the CALIBRATION field. The CALIBRATION field allows the clock to be adjusted in software, rather than via a hardware trimming capacitor as in most systems. When S is one, the CALIBRATION field value increases the oscillator frequency; when S is zero, it decreases the oscillator frequency. Calibration takes place over a 64 minute cycle. The first 62 minutes in the cycle may, once per minute, have one second shortened by 128 or lengthened by 256 oscillator cycles. The value of CALIBRATION determines how many minutes get an adjusted second. Each calibration step has the effect of adding 512 or subtracting 256 oscillator cycles for every 125,829,120 actual oscillator cycles.

The clock calibration can be achieved via program control by trial and error, or by intelligent comparison against the CIA's software clock, driven by a regulated 50/60Hz. If more accuracy is desired, the aforementioned FT pin can be measured with a frequency counter.

Additional information on this part may be found in the spec sheet, "MK41T56N TimeKeeper™ RAM", from SGS-Thomson Microelectronics.

A.5 Additional Pandora Information

Some additional material on the Pandora chip set is reproduced here. Greater details are available from Commodore in other documents

A.5.1 Common Pandora Display Modes

This table indicates the system bandwidth necessary to support each of the display modes. Note that interlace in any of the modes is possible, and carries no bandwidth penalty, only a display-time penalty (eg, it'll flicker).

Mode	LORES	HIRES	SHIRES	VGA
1-2 bitplanes	x1	x1	x1	x1
3-4 bitplanes	x1	x1	x2	x2
5-8 bitplanes	x1	x2	x4	x4
6 bitplane EHB	x1	x2	x4	x4
6 bitplane HAM	x1	x2	x4	x4
8 bitplane HAM	x1	x2	x4	x4
Dual Playfield, 2 x 2 bitplanes	x1	x1	x2	x2
Dual Playfield, 2 x 4 bitplanes	x1	x2	x4	x4

A.5.2 Pandora Register Summary

The following is a list of the Pandora registers and a brief description of what each one does. Each register is listed by name. The type is coded; "&" indicates the register is used by DMA channel only, "%" indicates the register is used by DMA channel usually, processors sometimes. Address register pairs are indicated by "+", while registers in the copper-protected area are indicated with a "~" next to their address. Chip family is indicated by an "h" for ECS, a "p" for Pandora. Registers are marked "R" for read, "W" for write, "ER" for early read, or "S" for strobe. The register's physical location is given by "A" for Agnus/Alice, a "D" for Denise/Lisa, and a "P" for Paula.

Name	Type	Addr	R/W	Chip(s)	Function
BLTDDAT	&	~000	ER	A	Blitter dest. early read (dummy address)
DMACONR		~002	R	A P	DMA control (and blitter status) read
VPOSR		~004	R	A	Read Vert most sig. bits (and frame flop)
VHPOSR		~006	R	A	Read Vert and horiz Position of beam
DSKDATR	&	~008	ER	P	Disk data early read (dummy address)
JOY0DAT		~00A	R	D	Joystick-mouse 0 data (vert,horiz)
JOY1DAT		~00C	R	D	Joystick-mouse 1 data (vert,horiz)

Name	Type	Addr	R/W	Chip(s)	Function
CLXDAT		~00E	R	D	Collision data reg.(Read and clear)
ADKCONR		~010	R	P	Audio, disk control register read
POT0DAT		~012	R	P	Pot counter pair 0 data (vert,horiz)
POT1DAT		~014	R	P	Pot counter pair 1 data (vert,horiz)
POTINP		~016	R	P	Pot pin data read
SERDATR		~018	R	P	Serial Port Data and Status read
DSKBYTR		~01A	R	P	Disk Data byte and status read
INTENAR		~01C	R	P	Interrupt Enable bits read
INTREQR		~01E	R	P	Interrupt Request bits read
DSKPTH	+	~020	W	A	Disk pointer, MSW
DSKPTL	+	~022	W	A	Disk pointer, LSW
DSKLEN		~024	W	P	Disk length
DSKDAT	&	~026	W	P	Disk DMA Data write
REFPTR	&	~028	W	A	Refresh pointer
VPOSW		~02A	W	A	Write Vert most sig. bits (and frame flop)
VHPOSW		~02C	W	A	Write Vert and horiz Position of beam
COPCON		~02E	W	A	Coprocessor control register (CDANG)
SERDAT		~030	W	P	Serial Port Data and stop bits write
SERPER		~032	W	P	Serial Port Period and control
POTGO		~034	W	P	Pot count start,pot pin drive enable and data
JOYTEST		~036	W	D	Write to all 4 Joystick-mouse counters at once.
STREQU	&	~038	S	D	Strobe for horiz sync with VB and EQU
STRVBL	&	~03A	S	D	Strobe for horiz sync with VB (Vert. Blank)
STRHOR	&	~03C	S	D P	Strobe for horiz sync
STRLONG	&	~03E	S	D	Strobe for identification of long horz. line.
BLTCON		~040	W	A	Blitter control register 0
BLTCON1		~042	W	A	Blitter control register 1
BLTAFWM		~044	W	A	Blitter first word mask for source A
BLTALWM		~046	W	A	Blitter last word mask for source A
BLTCPTH	+	~048	W	A	Blitter Pointer to source C, MSW
BLTCPTL	+	~04A	W	A	Blitter Pointer to source C, LSW
BLTBPTH	+	~04C	W	A	Blitter pointer to source B, MSW
BLTBPTL	+	~04E	W	A	Blitter pointer to source B, LSW
BLTAPTH	+	~050	W	A	Blitter Pointer to source A, MSW
BLTAPTL	+	~052	W	A	Blitter Pointer to source A, LSW
BLTDPTH	+	~054	W	A	Blitter Pointer to destn. D, MSW
BLTDPTL	+	~056	W	A	Blitter Pointer to destn. D, LSW
BLTSIZE		~058	W	A	Blitter start and size (window width, height)
BLTCON0L	h	~05A	W	A	Blitter control 0 lower 8 bits (minterms)
BLTSIZV	h	~05C	W	A	Blitter V size (for 15 bit vert size)
BLTSIZH	h	~05E	W	A	Blitter H size & start (for 11 bit H size)
BLTCMOD		~060	W	A	Blitter Modulo for source C
BLTBMOD		~062	W	A	Blitter Modulo for source B

Name	Type	Addr	R/W	Chip(s)			Function
BLTAMOD		~064	W	A			Blitter Modulo for source A
BLTDMOD		~066	W	A			Blitter Modulo for destn. D
BLTCDAT	&	~070	W	A			Blitter source C data reg
BLTBDAT	&	~072	W	A			Blitter source B data reg
BLTADAT	&	~074	W	A			Blitter source A data reg
SPRHDAT	&h	078	W	A			ext logic UHRES sprite pointer & data identifier
LISAID	h	~07C	R		D		Chip Revision level for Denise/Lisa
DSKSYNC		~07E	W			P	Disk sync pattern register for disk read.
COP1LCH	+	080	W	A			Coprocessor first location reg, MSW
COP1LCL	+	082	W	A			Coprocessor first location reg., LSW
COP2LCH	+	084	W	A			Coprocessor second location reg., MSW
COP2LCL	+	086	W	A			Coprocessor second location reg., LSW
COPJMP1		088	S	A			Coprocessor restart at first location
COPJMP2		08A	S	A			Coprocessor restart at second location
COPINS		08C	W	A			Coprocessor inst. fetch identify
DIWSTRT		08E	W	A	D		Display Window Start (upper left vert-hor pos)
DIWSTOP		090	W	A	D		Display Window Stop (lower right vert-hor pos)
DDFSTRT		092	W	A			Display bit plane data fetch start(hor pos)
DDFSTOP		094	W	A			Display bit plane data fetch stop(hor pos)
DMACON		096	W	A		P	DMA control write(clear or set)
CLXCON		098	W		D		Collision control
INTENA		09A	W			P	Interrupt Enable bits (clear or set bits)
INTREQ		09C	W			P	Interrupt Request bits (clear or set bits)
ADKCON		09E	W			P	Audio, Disk, UART, Control
AUD0LCH	+	0A0	W	A			Audio channel 0 location, MSW
AUD0LCL	+	0A2	W	A			Audio channel 0 location, LSW
AUD0LEN		0A4	W			P	Audio Channel 0 length
AUD0PER		0A6	W			P	Audio channel 0 Period
AUD0VOL		0A8	W			P	Audio Channel 0 Volume
AUD0DAT	&	0AA	W			P	Audio channel 0 Data
AUD1LCH	+	0B0	W	A			Audio channel 1 location, MSW
AUD1LCL	+	0B2	W	A			Audio channel 1 location, LSW
AUD1LEN		0B4	W			P	Audio Channel 1 length
AUD1PER		0B6	W			P	Audio channel 1 Period
AUD1VOL		0B8	W			P	Audio Channel 1 Volume
AUD1DAT	&	0BA	W			P	Audio channel 1 Data
AUD2LCH	+	0C0	W	A			Audio channel 2 location, MSW
AUD2LCL	+	0C2	W	A			Audio channel 2 location, LSW
AUD2LEN		0C4	W			P	Audio Channel 2 length
AUD2PER		0C6	W			P	Audio channel 2 Period
AUD2VOL		0C8	W			P	Audio Channel 2 Volume
AUD2DAT	&	0CA	W			P	Audio channel 2 Data
AUD3LCH	+	0D0	W	A			Audio channel 3 location, MSW

Name	Type	Addr	R/W	Chip(s)	Function
AUD3LCL	+	0D2	W	A	Audio channel 3 location, LSW
AUD3LEN		0D4	W	P	Audio Channel 3 length
AUD3PER		0D6	W	P	Audio channel 3 Period
AUD3VOL		0D8	W	P	Audio Channel 3 Volume
AUD3DAT	&	0DA	W	P	Audio channel 3 Data
BPL1PTH	+	0E0	W	A	Bit plane 1 pointer, MSW
BPL1PTL	+	0E2	W	A	Bit plane 1 pointer, LSW
BPL2PTH	+	0E4	W	A	Bit plane 2 pointer, MSW
BPL2PTL	+	0E6	W	A	Bit plane 2 pointer, LSW
BPL3PTH	+	0E8	W	A	Bit plane 3 pointer, MSW
BPL3PTL	+	0EA	W	A	Bit plane 3 pointer, LSW
BPL4PTH	+	0EC	W	A	Bit plane 4 pointer, MSW
BPL4PTL	+	0EE	W	A	Bit plane 4 pointer, LSW
BPL5PTH	+	0F0	W	A	Bit plane 5 pointer, MSW
BPL5PTL	+	0F2	W	A	Bit plane 5 pointer, LSW
BPL6PTH	+	0F4	W	A	Bit plane 6 pointer, MSW
BPL6PTL	+	0F6	W	A	Bit plane 6 pointer, LSW
BPL7PTH	+	0F8	W	A	Bit plane 7 pointer, MSW
BPL7PTL	+	0FA	W	A	Bit plane 7 pointer, LSW
BPL8PTH	+	0FC	W	A	Bit plane 8 pointer, MSW
BPL8PTL	+	0FE	W	A	Bit plane 8 pointer, LSW
BPLCON0		100	W	A D	Bit plane control register(misc control bits)
BPLCON1		102	W	D	Bit plane control reg (scroll value PF1, PF2)
BPLCON2		104	W	D	Bit plane control reg (priority control)
BPLCON3		106	W	D	Bit plane control reg (enhanced features)
BPL1MOD		108	W	A	Bit plane modulo (odd planes,or active-fetch lines if bitplane scan-doubling enabled)
BPL2MOD		10A	W	A	Bit Plane modulo (even planes or inactive-fetch lines if bitplane scan-doubling enabled)
BPLCON4	p	10C	W	D	Bit plane control reg (bitplane and sprite masks)
CLXCON2	p	10E	W	D	Extended collision control register
BPL1DAT	&	110	W	D	Bit plane 1 data (Parallel to serial convert)
BPL2DAT	&	112	W	D	Bit plane 2 data (Parallel to serial convert)
BPL3DAT	&	114	W	D	Bit plane 3 data (Parallel to serial convert)
BPL4DAT	&	116	W	D	Bit plane 4 data (Parallel to serial convert)
BPL5DAT	&	118	W	D	Bit plane 5 data (Parallel to serial convert)
BPL6DAT	&	11A	W	D	Bit plane 6 data (Parallel to serial convert)
BPL7DAT	&p	11C	W	D	Bit plane 7 data (Parallel to serial convert)
BPL8DAT	&p	11E	W	D	Bit plane 8 data (Parallel to serial convert)
SPR0PTH	+	120	W	A	Sprite 0 pointer, MSW
SPR0PTL	+	122	W	A	Sprite 0 pointer, LSW
SPR1PTH	+	124	W	A	Sprite 1 pointer, MSW
SPR1PTL	+	126	W	A	Sprite 1 pointer, LSW

Name	Type	Addr	R/W	Chip(s)	Function
SPR2PTH	+	128	W	A	Sprite 2 pointer, MSW
SPR2PTL	+	12A	W	A	Sprite 2 pointer, LSW
SPR3PTH	+	12C	W	A	Sprite 3 pointer, MSW
SPR3PTL	+	12E	W	A	Sprite 3 pointer, LSW
SPR4PTH	+	130	W	A	Sprite 4 pointer, MSW
SPR4PTL	+	132	W	A	Sprite 4 pointer, LSW
SPR5PTH	+	134	W	A	Sprite 5 pointer, MSW
SPR5PTL	+	136	W	A	Sprite 5 pointer, LSW
SPR6PTH	+	138	W	A	Sprite 6 pointer, MSW
SPR6PTL	+	13A	W	A	Sprite 6 pointer, LSW
SPR7PTH	+	13C	W	A	Sprite 7 pointer, MSW
SPR7PTL	+	13E	W	A	Sprite 7 pointer, LSW
SPR0POS	%	140	W	A D	Sprite 0 Vert-Horiz start position data
SPR0CTL	%	142	W	A D	Sprite 0 position and control data
SPR0DATA	%	144	W	D	Sprite 0 image data register A
SPR0DATB	%	146	W	D	Sprite 0 image data register B
SPR1POS	%	148	W	A D	Sprite 1 Vert-Horiz start position data
SPR1CTL	%	14A	W	A D	Sprite 1 position and control data
SPR1DATA	%	14C	W	D	Sprite 1 image data register A
SPR1DATB	%	14E	W	D	Sprite 1 image data register B
SPR2POS	%	150	W	A D	Sprite 2 Vert-Horiz start position data
SPR2CTL	%	152	W	A D	Sprite 2 position and control data
SPR2DATA	%	154	W	D	Sprite 2 image data register A
SPR2DATB	%	156	W	D	Sprite 2 image data register B
SPR3POS	%	158	W	A D	Sprite 3 Vert-Horiz start position data
SPR3CTL	%	15A	W	A D	Sprite 3 position and control data
SPR3DATA	%	15C	W	D	Sprite 3 image data register A
SPR3DATB	%	15E	W	D	Sprite 3 image data register B
SPR4POS	%	160	W	A D	Sprite 4 Vert-Horiz start position data
SPR4CTL	%	162	W	A D	Sprite 4 position and control data
SPR4DATA	%	164	W	D	Sprite 4 image data register A
SPR4DATB	%	166	W	D	Sprite 4 image data register B
SPR5POS	%	168	W	A D	Sprite 5 Vert-Horiz start position data
SPR5CTL	%	16A	W	A D	Sprite 5 position and control data
SPR5DATA	%	16C	W	D	Sprite 5 image data register A
SPR5DATB	%	16E	W	D	Sprite 5 image data register B
SPR6POS	%	170	W	A D	Sprite 6 Vert-Horiz start position data
SPR6CTL	%	172	W	A D	Sprite 6 position and control data
SPR6DATA	%	174	W	D	Sprite 6 image data register A
SPR6DATB	%	176	W	D	Sprite 6 image data register B
SPR7POS	%	178	W	A D	Sprite 7 Vert-Horiz start position data
SPR7CTL	%	17A	W	A D	Sprite 7 position and control data
SPR7DATA	%	17C	W	D	Sprite 7 image data register A

Name	Type	Addr	R/W	Chip(s)	Function
SPR7DATB	%	17E	W	D	Sprite 7 image data register B
COLOR00		180	W	D	Color table 00
COLOR01		182	W	D	Color table 01
COLOR02		184	W	D	Color table 02
COLOR03		186	W	D	Color table 03
COLOR04		188	W	D	Color table 04
COLOR05		18A	W	D	Color table 05
COLOR06		18C	W	D	Color table 06
COLOR07		18E	W	D	Color table 07
COLOR08		190	W	D	Color table 08
COLOR09		192	W	D	Color table 09
COLOR10		194	W	D	Color table 10
COLOR11		196	W	D	Color table 11
COLOR12		198	W	D	Color table 12
COLOR13		19A	W	D	Color table 13
COLOR14		19C	W	D	Color table 14
COLOR15		19E	W	D	Color table 15
COLOR16		1A0	W	D	Color table 16
COLOR17		1A2	W	D	Color table 17
COLOR18		1A4	W	D	Color table 18
COLOR19		1A6	W	D	Color table 19
COLOR20		1A8	W	D	Color table 20
COLOR21		1AA	W	D	Color table 21
COLOR22		1AC	W	D	Color table 22
COLOR23		1AE	W	D	Color table 23
COLOR24		1B0	W	D	Color table 24
COLOR25		1B2	W	D	Color table 25
COLOR26		1B4	W	D	Color table 26
COLOR27		1B6	W	D	Color table 27
COLOR28		1B8	W	D	Color table 28
COLOR29		1BA	W	D	Color table 29
COLOR30		1BC	W	D	Color table 30
COLOR31		1BE	W	D	Color table 31
HTOTAL	h	1C0	W	A	Highest number count in horiz line (VARBEAMEN=1)
HSSTOP	h	1C2	W	A	Horiz line pos for HSYNC stop
HBSTRT	h	1C4	W	A	Horiz line pos for HBLANK start
HBSTOP	h	1C6	W	A	Horiz line pos for HBLANK stop
VTOTAL	h	1C8	W	A	Highest numbered Vertical line (VARBEAMEN=1)
VSSTOP	h	1CA	W	A	Vert. line pos for VSYNC stop
VBSTRT	h	1CC	W	A	Vert line for VBLANK start
VBSTOP	h	1CE	W	A	Vert line for VBLANK stop

Name	Type	Addr	R/W	Chip(s)	Function
SPRHSTRT	h	1D0	W	A	UHRES sprite vertical start
SPRHSTOP	h	1D2	W	A	UHRES sprite vertical stop
BPLHSTRT	h	1D4	W	A	UHRES bit plane vertical start
BPLHSTOP	h	1D6	W	A	UHRES bit plane vertical stop
HHPOSW	h	1D8	W	A	DUAL mode hires H beam counter write
HHPOSR	h	1DA	R	A	DUAL mode hires H beam counter read
BEAMCON0	h	1DC	W	A	Beam counter control register (SHRES,UHRES,PAL)
HSSTRT	h	1DE	W	A	Horizontal Sync start (VARHSY)
VSSTRT	h	1E0	W	A	Vertical Sync start (VARVSY)
HCENTER	h	1E2	W	A	Horizontal position for Vsync on interlace
DIWHIGH	h	1E4	W	A D	Display window- upper bits for start,stop
BPLHMOD	h	1E6	W	A	UHRES bit plane modulo
SPRHPTH	+h	1E8	W	A	UHRES sprite pointer, MSW
SPRHPTL	+h	1EA	W	A	UHRES sprite pointer, LSW
BPLHPTH	+h	1EC	W	A	VRam (UHRES) bit plane pointer, MSW
BPLHPTL	+h	1EE	W	A	VRam (UHRES) bit plane pointer, LSW
FMODE	p	1FC	W	A D	Fetch MODE register
NO-OP(NULL)		1FE			Can also indicate last 2 or 3 refresh cycles or the restart of the COPPER after lockup.

.

A.6 Additional CODEC Information

Some additional details on the CODEC chips are provided here. This should be sufficient information for basic programming of the devices, though in most cases, this won't be necessary, as the hardware details of each device is expected to be abstracted by a VCOS device driver. More detailed information on these devices will be provided as data sheets become available.

A.6.1 The Phone-Line CODEC

The format of the Phone-Line CODEC control registers are given in detail here. Many of the clock generation features are not currently of any importance to the A3000+.

Control Register 0

Bit(s)	Name	Explanation	Value	Effect
9	INTEN	Interpolation filter	0	Disabled
			1	Enabled
8-5	TS	TSYNC Rate	0	9600Hz
			1	8000Hz
			2	7200Hz
			3	4800Hz
			4	2400Hz
			5	1200Hz
			6	600Hz
			7	19,200Hz
			8	14,400Hz
			10	12,000Hz
4-3	SR	Sampling Rater	0	9.6kHz
			1	8.0kHz
			2	7.2kHz
2-0	OP	Clock Operating Modes	0	Asynchronous fallback
			4	V.32 TSYNC
			5	V.32 Internal Sync
			6	V.32 Loopback
			7	Async. fallback TSYNC

Control Register 1

Bits(s)	Name	Explanation	Value	Effect
9	SA87	Scale sampling 8/7	0	Disabled
			1	Enabled
4	PDA*	Analog Power-Down	0	Power-down
			1	Normal operation
3	PDD*	Digital Power-Down	0	Power-down
			1	Normal operation

2-0	FB	Filter bypass	0	No bypass
			2	ADC hi-pass filter
			3	ADC hi/lo-pass filter

Control Register 2: Receive Control

Bits(s)	Name	Explanation	Value	Effect
6-4	BA	Receive baud clock	0	2400
			1	1600
			2	1200
			3	600
3-0	BI	Receive bit-rate	0	9600
			1	8000
			2	7200
			3	4800
			4	2400
			5	1200
			6	600
			7	19,200
			8	14,400
			9	12,000
			10	19,200 unscaled by SA87

Control Register 3: Transmit Control

Bits(s)	Name	Explanation	Value	Effect
6-4	BA	Transmit baud clock	0	2400
			1	1600
			2	1200
			3	600
3-0	BI	Transmit bit-rate	0	9600
			1	8000
			2	7200
			3	4800
			4	2400
			5	1200
			6	600
			7	19,200
			8	14,400
			9	12,000
			10	19,200 unscaled by SA87

Control Register 4: ADC Phase Adjust

Bits(s)	Name	Explanation	Value	Effect
8	PHD	Phase Control	0	Phase advance
			1	Phase retard
7-0	PA	ADC Phase Adjust		

Control Register 5: DAC Phase Adjust

Bits(s)	Name	Explanation	Value	Effect
8	PHD	Phase Control	0	Phase advance
			1	Phase retard
7-0	PA	DAC Phase Adjust		

A.6.2 The Hi-Fi Audio CODEC

The format of the Hi-Fi Audio CODEC frames are given in detail here. The A3000+ expects the DSP3210 to supply the serial clocks during control frames, the CODEC to supply the serial clocks during data frames. All unused bits should be read or get written zero.

A.6.2.1 The Control Frame

The control frame is logically broken up into eight byte-wide logical registers. The bytes are arranged from slot 0 to slot 7, MSB first, to create the control frame.

Control Slot 0: Status Register

Bits(s)	Name	Explanation	Value	Effect
2	DCB	Data control handshake		
0	AC	Force autocalibration	1	Write 1 to force

Control Slot 1: Data Format Register

Bits(s)	Name	Explanation	Value	Effect	
5-3	DFR	Data conversion frequency	0	8.00000kHz	5.51250kHz
		(depends on MCK)	1	16.00000kHz	11.02500kHz
			2	27.42857kHz	18.90000kHz
			3	32.00000kHz	22.05000kHz
			4	N/A	37.80000kHz
			5	N/A	44.10000kHz
			6	48.00000kHz	33.07500kHz
			7	9.60000kHz	6.61500kHz
2	ST	Mono/stereo	0	Mono	
			1	Stereo	
1-0	DF	Data format	0	16-bit linear	
			1	8-bit μ-law	
			2	8-bit A-law	
19	MCK	Master clock	0	Serial clock	
			1	XTAL1, 24.5760MHz	
			2	XTAL2, 16.9344MHz	

Control Slot 2: Serial Port Control Register

Bits(s)	Name	Explanation	Value	Effect
5-4	MCK	Master clock source	0	Serial clock
			1	XTAL1, 24.5760MHz
			2	XTAL2, 16.9344MHz
3-2	FSEL	Frame size select	0	64 bits/frame
			1	128 bits/frame
			2	256 bits/frame
1	XCLK	Transmit clock	0	Use external clocks
			1	Generate clocks
0	XEN	Transmitter enable	0	Enable serial output
			1	Disable serial output

Control Slot 3: Test Register

Bits(s)	Name	Explanation	Value	Effect
1	ADL	Loopback mode	0	Digital loopback mode
			1	Analog loopback mode
0	ENL	Loopback enable	0	Loopback disabled
			1	Loopback enabled

Control Slot 4: Parallel Port Register

Bits(s)	Name	Explanation	Value	Effect
7	PIO1	Parallel I/O 1		
6	PIO0	Parallel I/O 0		

C**ontrol Slot 6: Version Register**

Bits(s)	Name	Explanation	Value	Effect
3-0	RV	Device revision code	0	

A.6.2.2 The Data Frame

The data frame is logically broken up into eight byte-wide logical registers. The bytes are arranged from slot 0 to slot 7, MSB first, to create the data frame. For eight-bit frames, only the MSBs are used. For mono frames, only the Left channel is used.

Data Slot 0-1: Left Audio

Bits(s)	Name	Explanation	Value	Effect
15-0	LEFT	Left Channel Audio (MSB/LSB)		

Data Slot 2-3: Right Audio

Bits(s)	Name	Explanation	Value	Effect
15-0	RIGHT	Right Channel Audio		

Data Slot 4: Ouput Control

Bits(s)	Name	Explanation	Value	Effect
7	LE	Line output enable	0	Analog line output off
			1	Analog line output on
6	HE	Headphone output enable	0	Headphone output off
			1	Headphone output on
5-0	LO	Left channel attenuation	0	No attenuation
			62	93dB attenuation
			63	Digital mute

Data Slot 5: Ouput Control

Bits(s)	Name	Explanation	Value	Effect
6	SE	Speaker output enable	0	Speaker off
			1	Speaker on
5-0	RO	Right channel attenuation	0	No attenuation
			62	93dB attenuation
			63	Digital Mute

Data Slot 6: Input Control

Bits(s)	Name	Explanation	Value	Effect
7	PIO1	Parallel I/O 1		
6	PIO0	Parallel I/O 0		
5	OVR	Audio level overrange	0	Within range
			1	Over range
4	IS	Input select	0	Line level input
			1	Microphone input
3-0	LG	Left input gain	0	No gain
			15	22.5dB gain

Data Slot 7: Input Control

Bits(s)	Name	Explanation	Value	Effect
7-4	MA	Monitor path attenuation	0	No attenuation
			14	84dB attenuation
			15	Monitor path muted
3-0	RG	Right input gain	0	No gain
			15	22.5dB gain

A.6.2.3 Reset Conditions

The Audio CODEC is reset by the A3000+ DSPRST* signal. The control register parameters on powerup are given here.

Register	Value	Register	Value
MCK	0	LO	63
AC	0	RO	63
LE	0	RG	0
IS	0	OVR	0
SRE	0	DFR	0
DF	1	FSEL	2
XCLK	0	ENL	0
ADL	0	HE	0
DCB	1	LG	0
MA	15	PIO1	1
SE	0	PIO0	1
ST	0	XEN	1

A.7 References

MC68030 Enhanced 32 Bit Microprocessor User's Manual, Second Edition, Motorola Inc., number MC68030UM/AD REV1.

WE® DSP3210 Digital Signal Processor, Information Manual, Revision 1.8.1, AT&T Microelectronics.

COM20020 Universal Local Area Network Controller (ULANC), Standard Microsystems Corporation, Component Products Division.

WD33C93A SCSI Bus Interface Controller, Western Digital Corporation.

WD33C93B (SBIC) Enhanced SCSI Bus Interface Controller, Western Digital Corporation.

The I^2C Bus Specification, Phillips Components.

TDA8420: Hi-Fi Stereo Audio Processor; I^2C Bus, Phillips Components.

PCF8583: Clock Calendar with 256x8-Bit Static RAM, Phillips Components.

MK41T56N TimeKeeper™ RAM, SGS-Thomson Microelectronics.

The Zorro III Bus Specification, Revision 1.10, Commodore-Amiga, Inc.

The A3000 Local Bus Slot, Revision 2.00, Commodore-Amiga, Inc.